

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Extensions des réseaux de Pétri

de Pierpont, A

Award date:
2012

Awarding institution:
Université de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.



Facultés Universitaires Notre-Dame de la Paix, Namur
Faculté d'informatique
Année académique 2011-2012

EXTENSIONS DES RÉSEAUX DE PETRI

Astrid de Pierpont

Mémoire présenté en vue de l'obtention du grade académique de
master en sciences informatiques

Résumé

La vérification de la robustesse est une étape importante de l'analyse de systèmes informatiques. Une partie de cette analyse consiste à modéliser le système étudié afin de le tester virtuellement et ainsi détecter les erreurs potentielles.

Les réseaux de Petri sont un outil de modélisation graphique et matriciel inventé par Carl Adam Petri. Ces derniers permettent de représenter le fonctionnement des systèmes pour ensuite les analyser formellement. Cependant, leur conception initiale ne permet pas de modéliser des systèmes plus complexes, faisant par exemple intervenir des notions de temporalité ou d'interaction avec l'environnement externe. Pour faire face à ces différents manquements, de nombreuses extensions ont été inventées et greffées aux réseaux de Petri initiaux.

Ce travail réalise un état de l'art des principales extensions des réseaux de Petri. Il apparaîtra clairement que ces extensions rajoutent un grand pouvoir d'expressivité aux réseaux ordinaires, mais qu'elles peuvent aussi compliquer leur analyse.

MOTS CLES : Réseaux de Petri, extension, modélisation, vérification, propriétés, système

Abstract

Checking the robustness is an important step in the analysis of computer systems. Part of this analysis consists in modeling the system being studied in order to virtually test and thus detect potential errors.

Petri nets are a graphical modeling and mathematical tool invented by Carl Adam Petri. These nets are able to represent the operation of systems in order to analyze them formally. However, their original design does not allow modeling of more complex systems, such as systems that involve temporality or interaction with the external environment. To cope with these shortcomings, many extensions were invented and grafted to the original Petri nets.

This work achieves a review of the major extensions of Petri nets. It will be clear that these extensions add a great expressive power to ordinary networks, but can also complicate their analysis.

KEYWORDS: Petri Nets, extensions, modeling, checking, properties, system

Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.

Je voudrais tout d'abord adresser toute ma gratitude au promoteur de mon travail, Monsieur Jean-Paul Leclercq, pour l'aide, les recommandations et les précieux conseils qu'il m'a apportés lors des différents suivis et qui ont pu alimenter ma réflexion.

Je désire aussi remercier les membres de ma famille ainsi que mon mari pour leur soutien et relecture.

Table des matières

Introduction	1
CHAPITRE 1 : Concepts et notions préliminaires	2
1.1 Une modélisation mathématique	2
1.2 Concepts de base sur les réseaux de Petri.....	2
1.3 La limitation des réseaux de Petri ordinaires	5
CHAPITRE 2 : Les réseaux de Petri	7
2.1 Aspect statique	7
2.2 Aspect dynamique	8
2.3 Propriété des réseaux de Petri.....	10
2.4 Analyse des propriétés des réseaux de Petri par les graphes	15
2.5 Analyse des propriétés des réseaux par l'algèbre linéaire	20
2.6 Méthode de réduction.....	22
CHAPITRE 3 : Quelques extensions des réseaux de Petri	29
3.1 Réseau à arc inhibiteur (Inhibitor arc)	29
3.2 Réseau synchronisé (Synchronized Petri Nets).....	31
3.3 Réseau de Petri Prédicats/Transitions (Predicate/Transition).....	37
3.4 Réseau à files	39
3.5 Réseau de Petri à objet.....	41
3.6 Réseau de Petri flou	44
3.7 Conclusion.....	51
CHAPITRE 4 : Les réseaux colorés	53
4.1 Propriété manquante aux réseaux de Petri.....	53
4.2 Définition	54
4.3 Caractéristiques	62
4.4 Dépliage	62
CHAPITRE 5 : Les réseaux temporisés.....	66
5.1 Propriété manquante aux réseaux de Petri.....	66
5.2 Définition	67
5.3 Caractéristiques	74
CHAPITRE 6 : Les réseaux temporels	75
6.1 Propriété manquante aux réseaux de Petri.....	75
6.2 Définition	75

6.3	Caractéristiques	82
CHAPITRE 7 : Les réseaux stochastiques.....		83
7.1	Propriété manquante aux réseaux de Petri.....	83
7.2	Définition	85
Conclusion.....		87
Bibliographie		89

Table des figures

FIGURE 1 PLACE, TRANSITION, ARC.....	2
FIGURE 2 MARQUAGE INITIAL.....	3
FIGURE 3 MARQUAGE FINAL	3
FIGURE 4 RÉSEAU PLACES/TRANSITIONS TIRABLES ET NON-TIRABLES	4
FIGURE 5 REPRÉSENTATION GRAPHIQUE ET MATRICIELLE.....	5
FIGURE 6 HIÉRARCHIE DES EXTENSIONS DES RÉSEAUX DE PETRI	6
FIGURE 7 MODÉLISATION DU FLUX DU SANG	8
FIGURE 8 FRANCHISSABILITÉ DES TRANSITIONS.....	9
FIGURE 9 ILLUSTRATION D'UN TIF EFFECTIF	9
FIGURE 10 ILLUSTRATION D'UNE SÉQUENCE DE TRANSITIONS	10
FIGURE 11 ILLUSTRATION D'UN RÉSEAU BORNÉ ET NON-BORNÉ.....	12
FIGURE 12 ILLUSTRATION D'UN BLOCAGE	13
FIGURE 13 RÉSEAU VIVANT	14
FIGURE 14 CONFLIT EFFECTIF OU NON	15
FIGURE 15 GRAPHE DES MARQUAGES.....	16
FIGURE 16 EXEMPLE D'ANALYSE	17
FIGURE 17 ILLUSTRATION D'UN GRAPHE DE COUVERTURE.....	18
FIGURE 18 EXEMPLE D'UN FRANCHISSEMENT DE TRANSITION.....	21
FIGURE 19 EXEMPLE D'UNE SÉQUENCE DE TRANSITIONS.....	21
FIGURE 20 EXEMPLE 1 DE RÉDUCTION $R1$	23
FIGURE 21 EXEMPLE 2 DE RÉDUCTION $R1$	24
FIGURE 22 EXEMPLE 3 DE RÉDUCTION $R1$	24
FIGURE 23 EXEMPLE 4 DE RÉDUCTION $R1$	24
FIGURE 24 EXEMPLE 5 DE RÉDUCTION $R1$	25
FIGURE 25 RÉDUCTION $R1$ AVEC MARQUAGE: UNE SEULE TRANSITION.....	25
FIGURE 26 RÉDUCTION $R1$ AVEC MARQUAGE: PLUSIEURS TRANSITIONS	25
FIGURE 27 RÉDUCTION $R2$: PLACE IMPLICITE	26
FIGURE 28 RÉDUCTION $R2$: VÉRIFICATION DES CONDITIONS.....	26
FIGURE 29 RÉDUCTION $R3$: TRANSITION NEUTRE	27
FIGURE 30 RÉDUCTION $R4$: TRANSITIONS IDENTIQUES	27
FIGURE 31 RÉDUCTION TOTALE D'UN RÉSEAU DE PETRI.....	28
FIGURE 32 L'ARC INHIBITEUR ET TRANSITION TIRABLE	29
FIGURE 33 TRANSFORMATION D'UN ARC INHIBITEUR EN PLACE COMPLÉMENTAIRE	30
FIGURE 34 ILLUSTRATION DE RÉSEAU SYNCHRONISÉ : TRAITEMENT DE TEXTE.....	32
FIGURE 35 EXEMPLE DE DEUX TRANSITIONS ASSOCIÉES AU MÊME ÉVÈNEMENT	32
FIGURE 36 ÉVÈNEMENT CONTINU	33
FIGURE 37 EXEMPLE DE TIR ITÉRÉ SUR OCCURRENCE D'UN ÉVÈNEMENT	34
FIGURE 38 ÉTUDE DE LA BORNITUDE ENTRE RÉSEAU AUTONOME ET RÉSEAU SYNCHRONISÉ	35
FIGURE 39 GRAPHE DE MARQUAGE ET DE COUVERTURE DES RÉSEAUX AUTONOME ET SYNCHRONISÉ	35
FIGURE 40 ÉTUDE DE LA VIVACITÉ ENTRE RÉSEAU AUTONOME ET RÉSEAU SYNCHRONISÉ.....	36
FIGURE 41 RÉSEAU DE PETRI PRÉDICATS/TRANSITIONS.....	38
FIGURE 42 RÉSEAU À FILES.....	40

FIGURE 43 RÉSEAU DE PETRI À OBJET	43
FIGURE 44 ENSEMBLES FLOUS	45
FIGURE 45 DÉFINITION DES FONCTIONS D'APPARTENANCES.....	46
FIGURE 46 CALCUL DES VALEURS FLOUES POUR CHAQUE VARIABLE AVEC LA FONCTION D'APPARTENANCE.....	47
FIGURE 47 CALCUL DE LA RÈGLE FLOUE DE SORTIE.....	48
FIGURE 48 ILLUSTRATION DES DEUX FORMES	48
FIGURE 49 ILLUSTRATION D'UNE RÈGLE FLOUE	49
FIGURE 50 RÉSEAU DE PETRI AVEC L'AJOUT D'UNE MÉTHODE FLOUE	49
FIGURE 51 ILLUSTRATION DE L'OPÉRATEUR OR	50
FIGURE 52 VÉRIFICATION DE LA CONTINUITÉ ET DE LA FIN.....	50
FIGURE 53 MODÉLISATION D'UN SYSTÈME DE CHARIOT.....	53
FIGURE 54 RASSEMBLEMENT DES DEUX RÉSEAUX DE PETRI ORDINAIRES.....	54
FIGURE 55 IDENTIFICATION DE JETON PAR UNE COULEUR	54
FIGURE 56 ASSOCIATION DES COULEURS AUX TRANSITIONS	55
FIGURE 57 RÉSEAU AVEC EXPRESSIONS DES ARCS.....	56
FIGURE 58 TRANSITION NON VALIDE D'UN RÉSEAU COLORÉ	57
FIGURE 59 TRANSITION VALIDE D'UN RÉSEAU COLORÉ	57
FIGURE 60 AJOUT DE PLACES DE SORTIE AU RÉSEAU COLORÉ PRÉCÉDENT	58
FIGURE 61 ATTRIBUTION DES FONCTIONS À L'ARC	59
FIGURE 62 RÉSEAU COLORÉ COMPLEXE	60
FIGURE 63 FRANCHISSEMENT D'UNE TRANSITION PAR RAPPORT À UNE COULEUR	61
FIGURE 64 COMPORTEMENT D'UN RÉSEAU COLORÉ	61
FIGURE 65 DUPLICATION DES PLACES.....	63
FIGURE 66 MARQUAGE INITIAL.....	63
FIGURE 67 DUPLICATION DES TRANSITIONS.....	64
FIGURE 68 JONCTIONS ENTRE LES PLACES ET LES TRANSITIONS.....	64
FIGURE 69 RÉSEAU COLORÉ AVANT DÉPLIAGE.....	65
FIGURE 70 DÉPLIAGE DU RÉSEAU COLORÉ.....	65
FIGURE 71 RÉSEAU ORDINAIRE D'UNE RECETTE D'ŒUF À LA COQUE	66
FIGURE 72 INSERTION DE LA DURÉE SUR UNE PLACE	66
FIGURE 73 LES DEUX MODES DE TEMPORISATIONS.....	67
FIGURE 74 FRANCHISSEMENT D'UNE TRANSITION TEMPORISÉE.....	67
FIGURE 75 P-TEMPORISATION	68
FIGURE 76 EXÉCUTION À VITESSE MAXIMALE.....	69
FIGURE 77 GRAPHE DES MARQUAGES À VITESSE MAXIMALE.....	69
FIGURE 78 CALCUL FRÉQUENCE DE FRANCHISSEMENT.....	70
FIGURE 79 FONCTIONNEMENT À VITESSE PROPRE	70
FIGURE 80 T-TEMPORISATION.....	71
FIGURE 81 EXÉCUTION À VITESSE MAXIMALE DANS RÉSEAU T-TEMPORISÉ.....	72
FIGURE 82 RÉSEAU T-TEMPORISÉ ET SON GRAPHE DES MARQUAGES À VITESSE MAXIMALE.....	72
FIGURE 83 RÉSEAU T-TEMPORISÉ ET SON GRAPHE DES MARQUAGES À VITESSE PROPRE.....	73
FIGURE 84 LE PASSAGE DU P VERS T-TEMPORISATION	73
FIGURE 85 LE PASSAGE DU T VERS P-TEMPORISATION	74
FIGURE 86 RÉSEAU DE PETRI DU COVOITURAGE.....	75
FIGURE 87 ILLUSTRATION D'UN RÉSEAU TEMPOREL	76

FIGURE 88 RÈGLE DE TIR DANS UN RÉSEAU TEMPOREL	77
FIGURE 89 RÉSEAU TEMPOREL À L'INSTANT $T=0$	77
FIGURE 90 RÉSEAU TEMPOREL À 8H15(ROUGE) ET 8H35(JAUNE)	78
FIGURE 91 RÉSEAU TEMPOREL À 8H45 (BLEU) ET À 9H15(VERT).....	78
FIGURE 92 ILLUSTRATION DES ÉTATS D'UN RÉSEAU TEMPOREL	79
FIGURE 93 RÉSEAU TEMPOREL ET SON GRAPHE DES CLASSES.....	81
FIGURE 94 TROIS RÉSEAUX TEMPORELS BORNÉS	82
FIGURE 95 RÉSEAU DE PETRI STOCHASTIQUE.....	85
FIGURE 96 GRAPHE DES MARQUAGES ACCESSIBLES	86

Introduction

Aujourd'hui, la robustesse des systèmes informatiques est devenue indispensable. En effet, l'ascension de ceux-ci, dans notre société, apporte un soutien à de multiples besoins. Il est donc essentiel que les systèmes informatiques fonctionnent sans défaillance et répondent exactement aux exigences de leurs utilisateurs. Cependant, certains systèmes connaissent encore des problèmes de conceptions et de vérification des spécifications qui les empêchent de satisfaire les exigences attendues, occasionnant des difficultés à l'environnement du système. Prenons l'exemple d'une application développée pour gérer une série de feux de signalisations qui ne répond pas correctement aux spécifications. Celle-ci peut menacer la sécurité routière si les feux ne fonctionnent pas correctement. Une des causes de ces problèmes résulte d'une mauvaise analyse des systèmes. L'analyse est une discipline qui demande de décrire précisément ce que l'on attend d'un système avant son développement. Elle permet ainsi aux concepteurs de vérifier la compréhension qu'ils ont d'un système avec leurs clients.

L'analyse comprend différentes phases et la modélisation en fait partie. Elle consiste à donner une description détaillée du fonctionnement d'un système. Les concepteurs obtiennent une représentation, par exemple graphique, de l'organisation du système. Grâce celle-ci, ils exécutent différentes simulations de fonctionnement. Ainsi, la robustesse d'un système est testée et des erreurs de modélisation sont identifiées dans le cas de dysfonctionnement.

Pour modéliser un système, différents outils existent. Cependant, ces derniers ne permettent pas toujours d'identifier clairement les erreurs de conceptions. Or, l'identification de ces erreurs est indispensable pour le bon fonctionnement du système. Cette détection est compliquée, mais peut être réalisée grâce des modélisations plus formelles. Ces dernières sont basées sur des principes mathématiques qui permettent d'apporter une preuve du bon fonctionnement de l'application. Des erreurs imperceptibles dans la représentation graphique peuvent être ainsi identifiées. Les réseaux de Petri proposent ce type de modélisation : ils offrent une représentation graphique du fonctionnement d'un système ainsi qu'une représentation matricielle de celui-ci. Les matrices permettent d'étudier le fonctionnement du système.

Cependant, les réseaux de Petri furent rapidement limités par leur conception même. Par exemple, ils ne tiennent pas compte des événements extérieurs ou du temps d'exécution des tâches. Certaines personnes commencèrent donc à greffer des extensions sur les réseaux de Petri pour combler ces lacunes. Chacune de celles-ci essaye d'apporter une solution à un problème de modélisation particulière. Cependant, peu d'analystes utilisent actuellement ces extensions, car elles restent relativement méconnues.

Le manque d'utilisation de ces extensions est probablement dû à un manque de visibilité de celles-ci. Elles sont en effet peu accessibles dans la littérature et peu d'ouvrages les regroupent. Ce travail a pour but de donner un état de l'art des principales extensions des réseaux de Petri. Certaines sont peut-être déjà relativement connues, alors que d'autres commencent à peine à faire leur apparition. Ces nouvelles extensions permettent aux réseaux de Petri d'évoluer dans le monde de la modélisation et la vérification des propriétés et spécifications des grands systèmes.

CHAPITRE 1 : Concepts et notions préliminaires

1.1 Une modélisation mathématique

Certains secteurs tels que l'industrie nucléaire, l'aérospatiale et la métallurgie requièrent un haut niveau de sécurité, de performance et de contrôle. Le développement d'outils informatiques permet de satisfaire ces exigences. Cependant, ceux-ci doivent être exempts d'erreurs et conçus de manière rigoureuse.

Pour répondre à cette demande, Monsieur Carl Adam Petri, mathématicien allemand et informaticien, introduit les réseaux de Petri en 1962 dans une partie de sa thèse de doctorat « *Kommunikation mit Automaten* ». Ceux-ci permettent de modéliser des systèmes de contrôle afin d'en assurer leur conception, leur évaluation et leur amélioration.

Dans les réseaux de Petri, la modélisation d'un système se fait par une représentation graphique donnant une vue d'ensemble des différents composants et de leurs interactions. Cette représentation est traduite en équations matricielles qui permettent de vérifier de manière rigoureuse et mathématique certaines caractéristiques désirables du fonctionnement du système. Cette vérification va, par exemple, identifier dans le réseau des éléments inutiles ou des processus temporellement finis qui s'exécutent indéfiniment.

1.2 Concepts de base sur les réseaux de Petri

Dans cette section, nous allons présenter les éléments de base qui constituent la représentation graphique des réseaux de Petri. Nous montrerons ensuite un aperçu de la représentation matricielle. Les réseaux de Petri seront présentés de manière plus complète et précise dans le chapitre 2.

1.2.1 Représentation graphique

Un réseau de Petri est un graphe orienté composé de deux types de nœuds (voir figure 1) :

- Les **places**, représentées par un cercle, permettent de modéliser les états du système.
- Les **transitions**, représentées par un trait ou un rectangle, modélisent les changements d'état ou les événements sur les états du système.

Ces nœuds sont reliés par des **arcs orientés**.

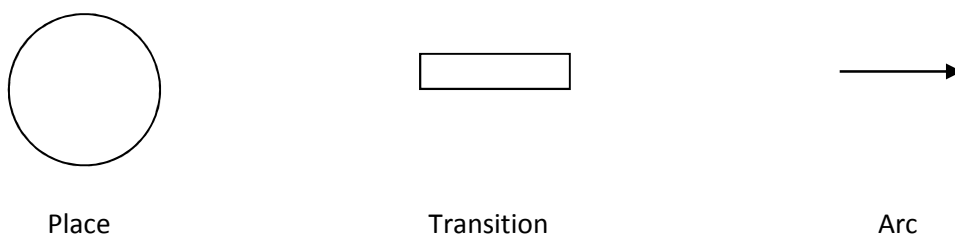


Figure 1 Place, transition, arc

Une place peut contenir un certain nombre de **jetons** ou **marques**. L'ensemble des marques, présentes à un instant donné dans les places, constitue le **marquage** du réseau à cet instant.

Une transition se compose d'un ou plusieurs arcs d'entrée et d'un ou de plusieurs arcs de sortie. Ces arcs permettent au réseau d'évoluer, c'est-à-dire de permettre aux jetons de se déplacer. (voir figure 2)

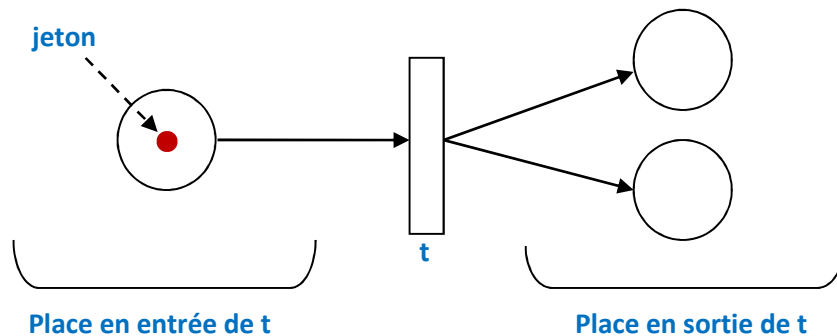


Figure 2 Marquage initial

On parle aussi de **préconditions** et de **postconditions** pour les places en entrée et en sortie par rapport à une transition ou une action. Une précondition est l'ensemble des places qui doivent être vraies pour que l'action, la transition, puisse être effectuée. Une place est vraie lorsqu'elle contient un jeton.

Une postcondition est l'ensemble des places qui sont vraies à l'issue du déroulement de l'action. Pour franchir une transition, le jeton présent dans une place d'entrée est supprimé de cette place et ensuite déposé dans chacune des places en sortie (voir figure 3).

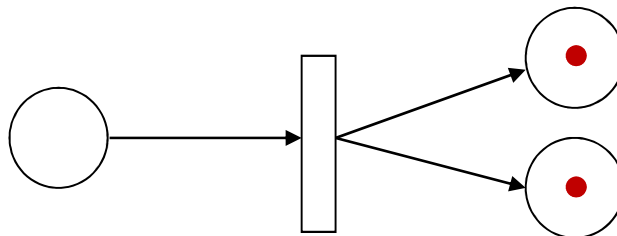


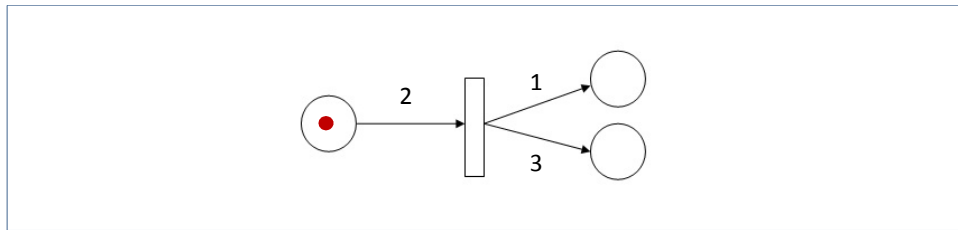
Figure 3 Marquage final

On parlera alors d'une transition **tirable** ou **franchissable**, si les conditions requises pour le déclenchement de l'action sont satisfaites. C'est-à-dire, si les préconditions et les postconditions sont valides.

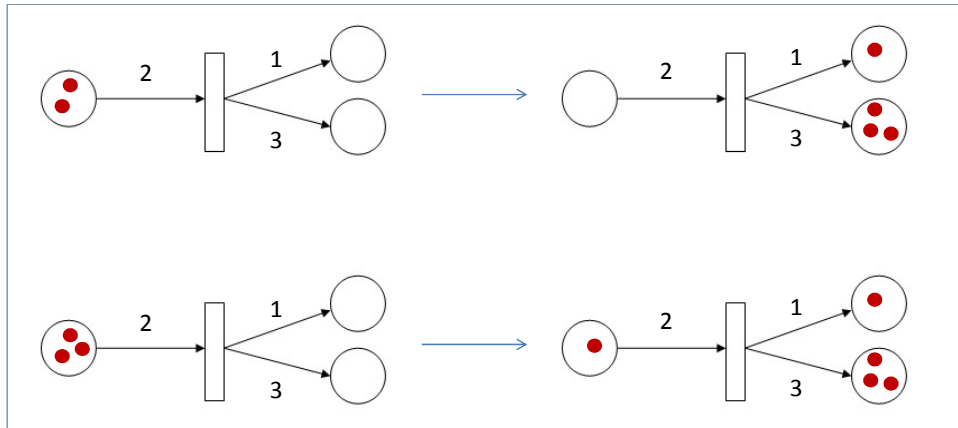
Ce déplacement va modifier le marquage du graphe. Celui-ci permet d'analyser le fonctionnement du réseau, c'est-à-dire le déroulement attendu du futur système.

Le réseau de Petri présenté ci-dessus est appelé réseau **conditions/événements**. Ce dernier utilise des conditions booléennes. Il existe un autre type de réseau basé sur le précédent, appelé **réseau places/transitions**. Celui-ci, par contre, utilise des conditions entières. Ces conditions se traduisent par la présence de poids sur les arcs. Ces poids vont déterminer le nombre de jetons qui doivent être présents dans une place en entrée afin que la transition puisse être tirée ou le nombre de jetons qui seront placés dans une place en sortie. Ceci implique qu'une place puisse

contenir plusieurs jetons. La place représente une ressource et un jeton présent dans cette place modélise une instance de cette ressource. Donc si plusieurs instances de ressources sont nécessaires, il y aura plusieurs jetons dans la place.



Transition non tirable car le nombre de jeton(s) dans la place en entrée < poids de l'arc en entrée



Transition tirable car le nombre de jeton(s) dans la place en entrée \geq poids de l'arc en entrée

Figure 4 Réseau places/transitions tirables et non-tirables

1.2.2 Représentation matricielle

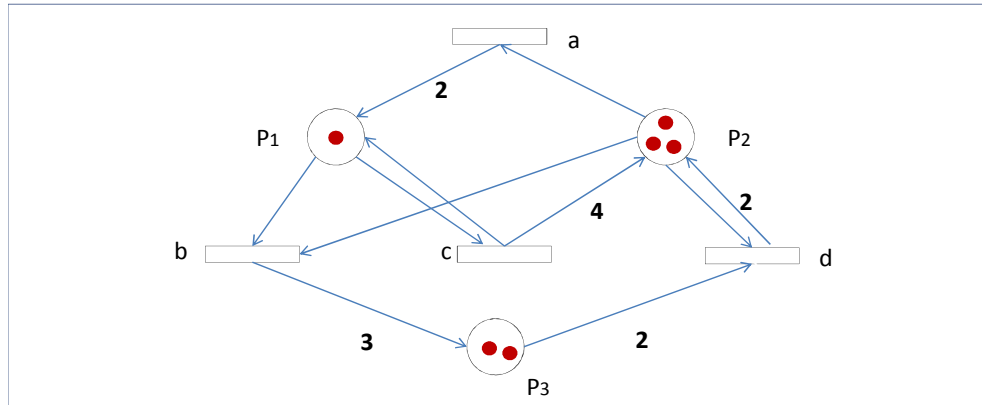
Après la réalisation de la représentation graphique, il est souhaitable d'étudier le fonctionnement du réseau. L'outil d'analyse utilisé pour cette étude est issu de l'algèbre linéaire. Trois matrices et un vecteur sont générés à partir de la représentation graphique :

- La matrice des préconditions, *Pré*, qui modélise les préconditions de tir des différentes transitions.
- La matrice des postconditions, *Post*, qui modélise les postconditions de tir des différentes transitions.
- La matrice d'incidence, *C*, se construit en faisant la différence entre la matrice des postconditions et la matrice des préconditions.
- Le vecteur contient le marquage initial du réseau.

Une matrice contient un nombre de lignes égal au nombre de places et un nombre de colonnes égal au nombre de transitions.

S'il n'y a pas d'arc qui joint une place à une transition ou inversement, la valeur est mise à 0.

S'il n'y a pas de poids sur l'arc, la valeur est mise à 1.



Exemple de représentation graphique d'un réseau de Pétri

$$\begin{aligned}
 \text{Pré} &= \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 2 \end{pmatrix} & \text{Post} &= \begin{pmatrix} 2 & 0 & 1 & 0 \\ 0 & 0 & 4 & 2 \\ 0 & 3 & 0 & 0 \end{pmatrix} & C &= \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & -1 & 4 & 1 \\ 0 & 3 & 0 & -2 \end{pmatrix} \\
 M &= \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}
 \end{aligned}$$

Représentation matricielle du même réseau

Figure 5 Représentation graphique et matricielle

1.3 La limitation des réseaux de Petri ordinaires

Les réseaux de Petri ordinaires sont puissants, mais montrent vite leurs limites lorsqu'il s'agit de modéliser des systèmes plus complexes avec des conditions plus riches. Par exemple, il est compliqué, dans un réseau de Petri de base, de modéliser les durées d'une action, les débuts et les fins d'exécutions, les identifications, les interventions du contexte extérieur...

Pour répondre à ce problème, des extensions sont rajoutées au réseau de base. Chacune d'elles apporte une solution à une limite ou contrainte. Grâce à celles-ci, le réseau pourra contenir des informations dans les jetons, des informations sur le contexte, des informations sur le temps et la durée.

Le schéma suivant présente les différentes extensions les plus connues et utilisées. La suite de ce travail présentera chacune d'elles avec leurs avantages et inconvénients, leurs utilisations avec leurs facilités et difficultés. Il est important de signaler que ce travail ne peut présenter toutes les extensions produites depuis la création des réseaux de Petri. Nous avons donc sélectionné celles qui nous semblent les plus pertinentes et les plus utiles pour la création de réseaux de Petri plus complexes.

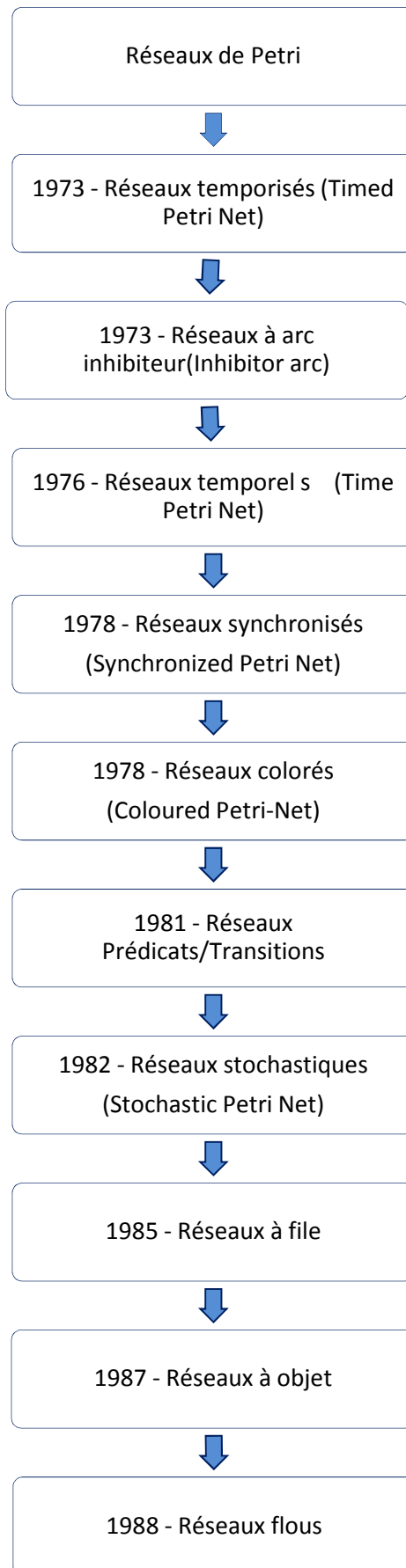


Figure 6 Hiérarchie des extensions des réseaux de Petri

CHAPITRE 2 : Les réseaux de Petri

2.1 Aspect statique

Les concepts de base des réseaux de Petri ont été donnés dans le chapitre 1, dans la section « Concepts de base sur les réseaux de Petri ». Toutefois, les réseaux de Petri étant des modèles formels, il convient d'en donner une définition rigoureuse.

Définition : Un **réseau de Petri** est représenté par un triplet $R = (P, T, W)$ où :

- P est un ensemble fini de places : $P = \{p_1, \dots, p_n\}$.
- T est un ensemble fini de transitions, disjoint de P : $T = \{t_1, \dots, t_m\}$.
- $W : P \times T \cup T \times P \rightarrow \mathbb{N}$ est la fonction de valuation des arcs.

La fonction de valuation W rassemble les préconditions et les postconditions. Elles sont définies de la manière suivante :

- $\text{Pré} : T \times P \rightarrow \mathbb{N}$ est la restriction de W à $P \times T$. Elle exprime les préconditions requises pour qu'une action soit réalisable. $\text{Pré}(t, p) = W(p, t)$.
- $\text{Post} : T \times P \rightarrow \mathbb{N}$ est la restriction de W à $T \times P$. Elle exprime les conséquences de l'exécution d'une action. $\text{Post}(t, p) = W(t, p)$.

Définition : On appelle **marquage** d'un réseau $R = (P, T, W)$ toute fonction $M : P \rightarrow \mathbb{N}$. On appelle **réseau marqué** tout couple (R, M_0) où R est un réseau et M_0 un marquage, appelé **marquage initial**.

Pour une place p de P , $M(p)$ représente le nombre de jetons (ou de marquages) contenus dans la place p . Le marquage permet donc de synthétiser les ressources du système et les conditions vérifiées.

Illustrons tous ces concepts par l'exemple suivant : le flux du sang dans le cœur.

Les veines caves collectent le sang de la circulation veineuse et l'envoient dans l'oreillette droite du cœur grâce à la phase de décontraction du myocarde. Le sang passe ensuite dans le ventricule droit lors de l'ouverture de la valvule tricuspide. Après la fermeture de la valvule tricuspide, le ventricule droit est contracté grâce à l'ouverture de la valvule sigmoïde, le sang est éjecté vers les poumons. Après chargement en oxygène, il revient dans le cœur en passant par l'oreillette gauche. Il coule ensuite dans le ventricule gauche à travers la valve mitrale. La valve mitrale se ferme pour augmenter la pression dans le ventricule gauche et lors de l'ouverture de la valve aortique, le sang est renvoyé dans le système sanguin.



À travers cette description, nous pouvons identifier les places et les transitions

Places	N° des places	Transitions	N° des transitions
Système sanguin	P_1	Décontraction du myocarde	T_1
Oreillette droite	P_2	Ouverture valvule tricuspide	T_2
Ventricule droit	P_3	Fermeture valvule tricuspide	T_3
Ventricule droit contracté	P_4	Ouverture valvule sigmoïde	T_4
Poumons	P_5	Chargement oxygène	T_5
Oreillette gauche	P_6	Ouverture valve mitrale	T_6
Ventricule gauche	P_7	Fermeture valve mitrale	T_7
Ventricule gauche contracté	P_8	Ouverture valve aortique	T_8

Avec les places et transitions identifiées, nous pouvons construire le réseau de Petri suivant.

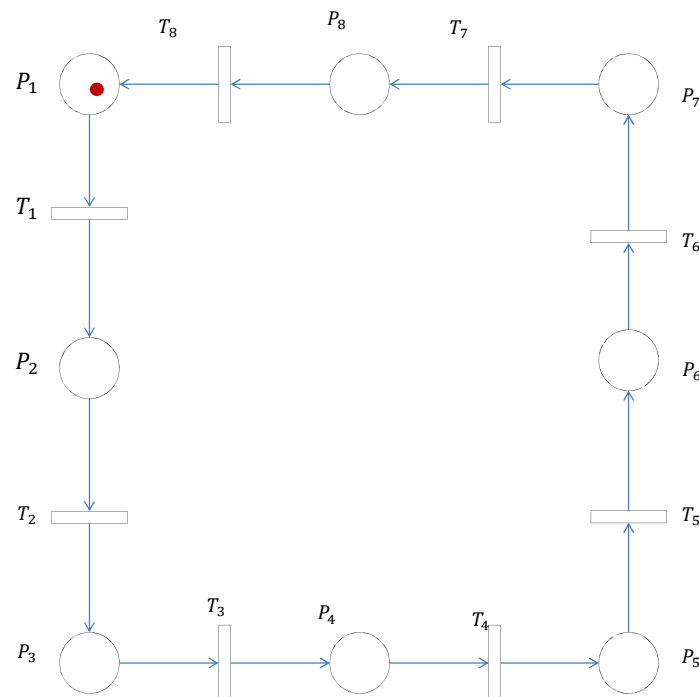


Figure 7 Modélisation du flux du sang

2.2 Aspect dynamique

Comme dit dans la section précédente, les concepts de base des réseaux de Petri ont été donnés dans le chapitre 1 dans la section « Concepts de base sur les réseaux de Petri ». Cette section présente et définit les éléments pour pouvoir réaliser une étude comportementale des réseaux de Petri. Cet aspect dynamique permet d'étudier l'évolution des systèmes dans le temps. Cette évolution est modélisée par la réalisation d'action et le passage par différents états. Si l'on souhaite analyser un système, cela suppose de modéliser son fonctionnement dans le temps. L'élément utilisé pour modéliser cet aspect est : le tir de transitions.

Définition : Soit N un réseau places/transitions, t une transition et M un marquage. Une transition t est **tirable** ou **sensibilisée** ou **franchissable** depuis le marquage M si et seulement si : $\forall p \in P, M(p) \geq W(p, t) (M \geq W(\blacksquare, t))$. On le note $M(t >)$.

Comme dit dans le chapitre 1, il faut vérifier que les ressources (jetons) nécessaires sont disponibles et que les conditions sont réalisables. La figure suivante illustre les conditions pour qu'une transition soit tirable.

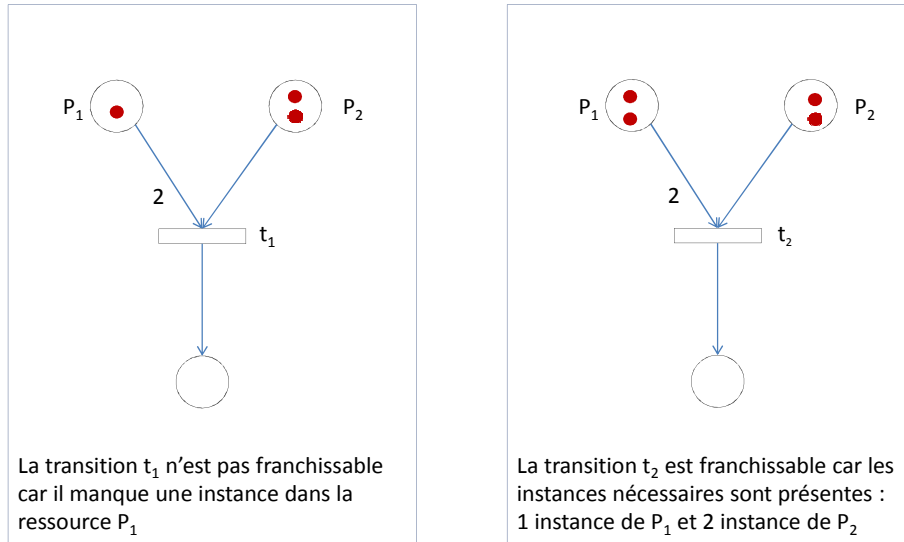


Figure 8 Franchissabilité des transitions

Comme on peut le voir, la tirabilité exprime une condition d'exécution d'une action. Le tir effectif d'une transition tirable modélise l'évolution du système.

Définition : Si $M(t >)$ alors le tir effectif de t conduit le système dans l'état décrit par le marquage M' défini par : $\forall p \in P, M'(p) = M(p) - W(p, t) + W(t, p) (M' = M - W(\blacksquare, t) + W(t, \blacksquare))$.

Que l'on note aussi $M(t > M')$.

Illustrons cette définition par un exemple. Nous considérons le réseau de la figure 9.

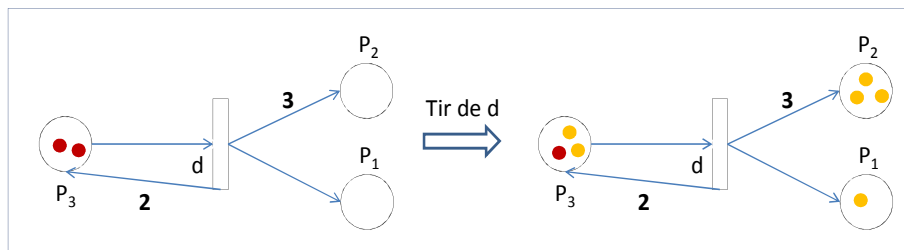


Figure 9 Illustration d'un tir effectif

Son marquage initial est $M_0 = (M_0(P_1), M_0(P_2), M_0(P_3)) = (0, 0, 2)$. On a alors $M_0(y > M = (1, 3, 3))$.

Grâce à l'ensemble des séquences des tirs de transitions, on modélise le fonctionnement et l'évolution du système.

Définition : Soit N un réseau places/transition, $\sigma = t_1 t_2 \dots t_n \in T^*$ une séquence de transitions et M un marquage. La séquence σ est franchissable depuis M si et seulement si il existe des marquages M_1, M_2, \dots, M_n tels que : $M(t_1 > M_1(t_2 > M_2(\dots M_{n-1}(t_n > M_n$. Le tir de σ conduit alors au marquage M_n . On le note $M(\sigma > M_n$.

Pour illustrer cette notion de séquence, prenons l'exemple à la figure 10. La séquence $\sigma = ded$ est franchissable depuis le marquage initial, et on a : $M_0 = (0,0,2)(\sigma > M = (1,4,4)$.

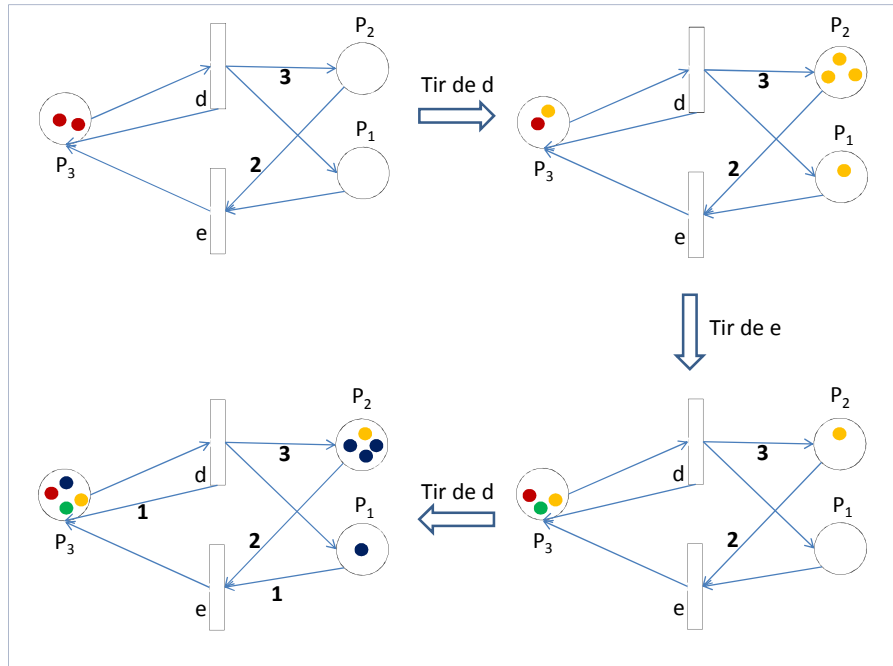


Figure 10 Illustration d'une séquence de transitions

2.3 Propriété des réseaux de Petri

Les réseaux de Petri permettent de modéliser des systèmes complexes. Ceux-ci se doivent donc de répondre à un certain nombre d'exigences (absence de blocage, préservation des fonctionnalités,...). Grâce aux réseaux de Petri, on peut vérifier un certain nombre de ces exigences, car elles sont liées à des propriétés du réseau. Les propriétés peuvent être liées soit à son état, soit à son évolution. Cette section explique et définit l'ensemble des propriétés qui sont exprimables par les réseaux de Petri.

2.3.1 Monotonie

La propriété de monotonie est liée à la franchissabilité des transitions. Une transition tirable à partir d'un marquage donné l'est également à partir d'un marquage plus grand. Cela implique que la franchissabilité s'appuie sur la présence d'un nombre minimum de ressources nécessaires à l'exécution d'une action.

Propriété : Soit N un réseau places/transitions. Soit t une transition et M un marquage tels que $M(t > M')$. Soit \tilde{M} tel que $M < \tilde{M}$ alors,

1. t est franchissable à partir de \tilde{M} : $\tilde{M}(t > ;$
2. Le marquage atteint est supérieur à M' : $\tilde{M}(t > \tilde{M}' \Rightarrow \tilde{M}' > M'$. De manière plus précise : $M(p) < \tilde{M}(p) \Leftrightarrow M'(p) < \tilde{M}'(p)$.

Cette propriété s'étend ensuite aux séquences de transitions. Une première application de cette propriété consiste en la caractérisation des séquences qui peuvent être itérées un nombre quelconque de fois.

Définition : Soit N un réseau places/transitions, σ une séquence de transitions et M et M' deux marquages tels que $M(\sigma > M')$. On dit que :

- σ est répétitive si et seulement si $M \leq M'$
- σ est répétitive stationnaire si et seulement si $M = M'$
- σ est répétitive croissante si et seulement si $M < M'$.
- $M < M'$: $M \leq M'$ et il existe $i : M(i) < M'(i)$

2.3.2 Terminaison finie ou infinie

Cette propriété définit la notion de régime permanent, c'est-à-dire un comportement infini. Cela se traduit par l'existence de séquence de transitions infinie franchissable depuis le marquage initial. Si cette séquence n'existe pas, on dit que le système est à terminaison finie : l'évolution du système se terminera toujours au bout d'un nombre fini d'actions.

Pour définir la notion de séquence infinie, nous devons définir le langage du réseau marqué. Le langage du réseau est l'ensemble de toutes les séquences de transitions franchissables depuis le marquage initial.

T^* représente l'ensemble des séquences σ constituées de transitions $t \in T$

T^∞ représente l'ensemble des séquences infinies σ constituées de transitions $t \in T$

Définition : une séquence répétitive est une séquence σ franchissable telle que chaque transition du réseau apparaît un nombre illimité de fois

Définition : L'ensemble des séquences tirables depuis le marquage M_0 soit $\{\sigma \in T^*/M_0(\sigma >)\}$ est le **langage** du réseau marqué $L(N, M_0)$

Définition : On dit qu'un réseau places/transitions marqué admet une **séquence infinie** σ si tout préfixe fini de σ est une séquence franchissable depuis M_0 : $\exists \sigma \in T^\infty / \forall \sigma' \in T^*, \sigma' \text{ préfixe de } \sigma \Rightarrow \sigma' \in L(N, M_0)$. Si une telle séquence n'existe pas, le système est dit à **terminaison finie**.

Cela veut simplement dire qu'il existe une séquence infinie σ si après un préfixe fini σ' , il existe une séquence répétitive infinie σ'' . σ est la composition de σ' et d'un nombre quelconque de fois σ'' .

Pour les séquences infinies, elles peuvent se caractériser de cette manière :

- Séquence répétitive stationnaire : séquence de franchissements σ tels que $M_0(\sigma) > M_0$.
- Séquence complète : séquence qui contient toutes les transitions du réseau de Petri.
- Séquence répétitive croissante : séquence de franchissements σ telle que $M_0(\sigma) > M'_0$ avec $M'_0 > M_0$
- Séquence répétitive décroissante : séquence de franchissements σ telle que $M_0(\sigma) > M''_0$ avec $M_0 > M''_0$

Une composante répétitive est l'ensemble T_0 des transitions de T apparaissant dans la séquence σ .

2.3.3 Réseaux et places bornés

Définition : Une place P_i est **bornée** pour un marquage initial M_0 si pour tout marquage accessible à partir de M_0 , le nombre de marques dans P_i reste borné. Elle est dite k -bornée si le nombre de marques dans P_i est toujours inférieur ou égal à k ($M'(p) \leq k$). Un réseau de Petri marqué est (k) -borné si toutes ses places sont (k) -bornées.

Il est important de remarquer que cette propriété dépend du marquage initial. Cela veut dire que si un réseau marqué n'est pas borné par un marquage initial, il peut être borné pour un autre marquage initial.

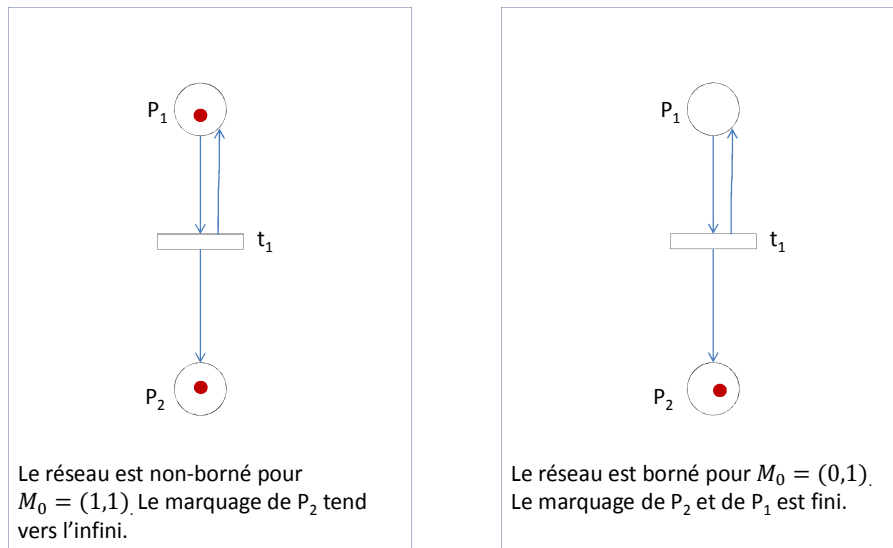


Figure 11 Illustration d'un réseau borné et non-borné

Cependant, si un réseau marqué n'est pas borné pour le marquage initial M_0 alors il n'est pas borné pour le marquage initial $M'_0 \geq M_0$.

De manière plus générale, un réseau est borné si toutes ses places restent bornées quel que soit son marquage initial M_0 .

Définition : Un réseau de Petri marqué est **sauf** ou **binaire** pour un marquage initial M_0 s'il est 1-borné.

2.3.4 Blocage et vivacité

Le franchissement de transitions permet l'évolution du marquage d'un réseau. Si, durant cette évolution, certaines transitions ne sont pas franchies, c'est-à-dire que l'évènement associé ne se produit pas, cela signifie que le marquage d'une partie du réseau n'évolue pas non plus. Cette constatation montre que le sous-système modélisé par cette partie du réseau ne fonctionnera pas. Il existe donc un problème dans la conception et il est important de pouvoir détecter ce type de problème.

2.3.4.1 Blocage

Le réseau est sans blocage si à partir de tout marquage accessible, il existe au moins une transition franchissable. Dans le cas contraire, tout marquage accessible à partir duquel on ne peut tirer aucune transition, est appelé marquage **mort**.

Définition : Un **blocage** (ou **état puits**) est un marquage pour lequel aucune transition n'est validée. Un réseau de Petri marqué est dit sans blocage pour un marquage initial M_0 si aucun marquage accessible est un blocage.

À la figure 12, on peut voir que le réseau, pour le marquage initial $M_0 = (0,1,1,2)$, a pour blocage le marquage $M_3 = (1,0,0,4)$

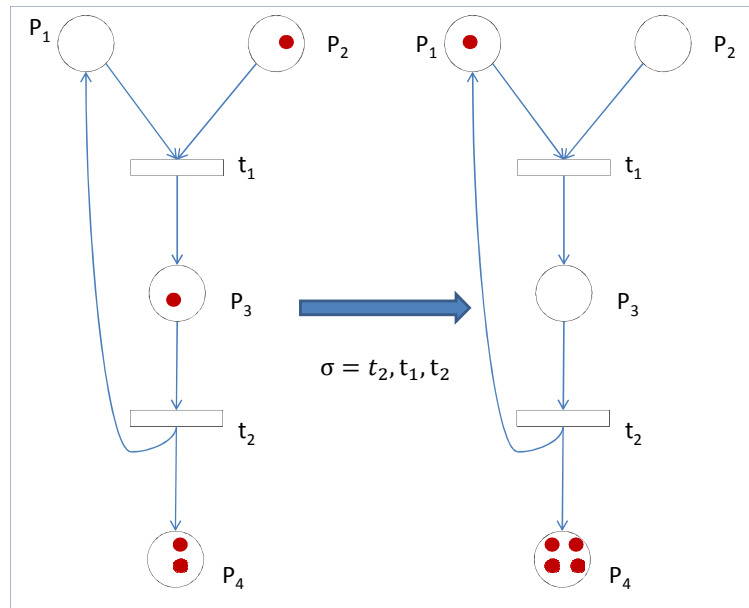


Figure 12 Illustration d'un blocage

Propriété : Si un réseau de Petri marqué est sans blocage pour un marquage initial M_0 alors il n'est pas nécessairement sans blocage pour $M'_0 \geq M_0$.

2.3.4.2 Quasi-vivant

On souhaite exprimer le fait que toutes les actions peuvent être exécutées au moins une fois. Dans le cas contraire, cela signifie qu'il existe des fonctionnalités du système auxquelles on n'a pas accès.

Définition : Une transition T_j est **quasi-vivante** pour un marquage initial M_0 s'il existe une séquence de franchissements à partir de M_0 contenant T_j : $\exists \sigma, M_0(\sigma >$ et $\sigma = \dots T_j \dots$

Propriété : Si la transition T_j est quasi-vivante pour un marquage initial M_0 alors elle est quasi-vivante pour $M'_0 \geq M_0$.

Le réseau de la figure 12 est quasi-vivant.

Définition : Un réseau de Petri marqué est **quasi-vivant** pour un marquage initial M_0 si toutes ses transitions sont quasi-vivantes pour ce marquage initial.

2.3.4.3 Vivant

Contrairement à la propriété « quasi-vivant », ici, cette propriété garantit qu'il n'y ait aucune action qui devienne définitivement inaccessible.

Définition : Une transition T_j est **vivante** pour un marquage initial M_0 si pour tout marquage accessible M_k , il existe une séquence de franchissements à partir de M_k contenant T_j :

$$\forall M_k \in * M_0, \exists \sigma, M_k(\sigma > \text{ et } \sigma = \dots T_j \dots$$

Si une transition T_j est vivante alors à tout instant, on sait que T_j peut être franchie dans le futur. Cela implique que le réseau, modélisant un système, fonctionne en permanence. Si une transition est non vivante, on peut traduire cette propriété par une erreur ou une panne dans le système.

Propriété : Si la transition T_j est vivante pour un marquage initial M_0 alors elle n'est pas nécessairement vivante pour $M'_0 \geq M_0$.

Définition : Un réseau de Petri est vivant pour un marquage initial M_0 si toutes ses transitions sont vivantes pour ce marquage initial. Un réseau de Petri est dit conforme s'il est sauf et vivant.

Le réseau de la figure 12 n'est pas vivant contrairement à celui de la figure 13.

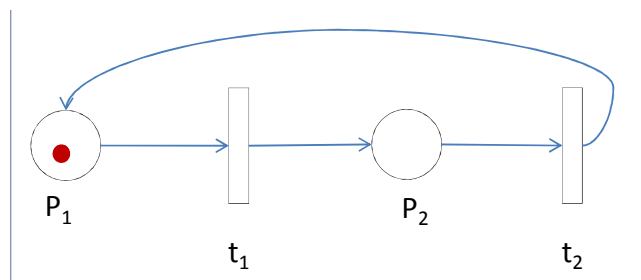


Figure 13 Réseau vivant

2.3.5 Conflits

Un conflit structurel est présent lorsqu'une place P_i a au moins **deux** transitions de sorties T_j et T_k . **Notation** : $\langle P_i, \{T_j, T_k, \dots\} \rangle$

Définition : Un conflit effectif est l'existence d'un conflit structurel $\langle P_i, \{T_j, T_k, \dots\} \rangle$ et d'un marquage M tel que le nombre de marques dans la place P_i est strictement inférieur au nombre de transitions de sortie de P_i validées par le marquage M .

Lorsqu'un conflit effectif se produit, il est nécessaire de choisir la transition qui doit être franchie.

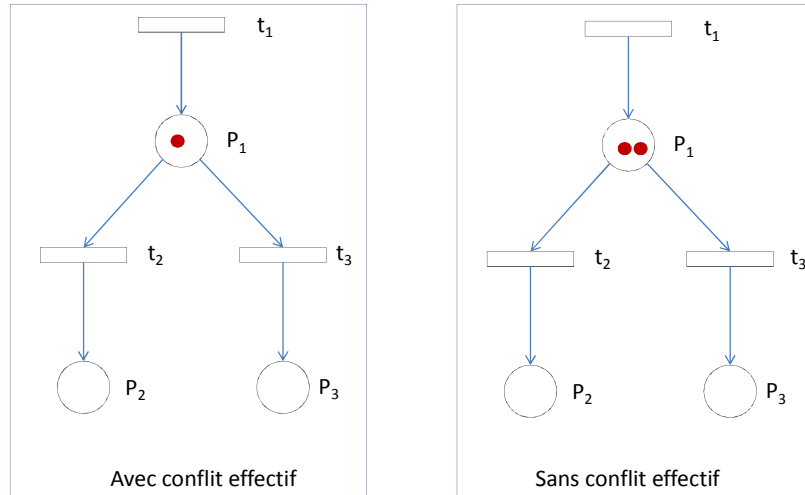


Figure 14 Conflit effectif ou non

Définition : Un réseau de Petri est persistant pour un marquage initial M_0 si pour tout marquage M_i accessible à partir de M_0 , on a : si T_j et T_k sont validées par le marquage M_i alors $T_j T_k$ est une séquence de franchissements à partir de M_i .

Si le réseau est persistant alors il n'est pas nécessaire d'effectuer un choix lors d'un conflit effectif.

Les sections suivantes expliquent comment étudier ces propriétés dans les réseaux de Petri.

2.4 Analyse des propriétés des réseaux de Petri par les graphes

Une des manières pour analyser les propriétés des réseaux de Petri est de construire le graphe de tous les marquages accessibles. Ce graphe est construit de la manière suivante : chaque sommet du graphe correspond à un marquage accessible depuis le marquage initial et chaque arc reliant deux sommets correspond au franchissement d'une transition permettant de passer d'un marquage à l'autre.

Si l'ensemble des marquages accessibles est fini, on utilise un graphe des marquages. Par contre, si l'ensemble des marquages accessibles est infini, on passe par un graphe de couverture. L'objectif de cette section est d'expliquer comment réaliser un graphe des marquages et un graphe de couverture et d'en extraire les différentes propriétés.

2.4.1 Graphe des marquages et analyse des propriétés

2.4.1.1 Graphe

Le graphe des marquages est un graphe orienté dont les sommets sont étiquetés par des marquages et dont les arcs sont étiquetés par des transitions.

Définition : On appelle **graphe des marquages** du réseau (N, M_0) le graphe étiqueté tel que :

- Le sommet initial est étiqueté par M_0
- Si v est un sommet d'étiquette M et si t est une transition telle que $M(t >$, avec $M(t > M'$, alors il existe un sommet v' d'étiquette M' et un arc

$$M \xrightarrow{t} M'$$

La méthode pour construire le graphe des marquages accessibles est très simple. Comme dit précédemment, les nœuds du graphe sont formés des différents marquages de M_0 . On commence par le marquage initial et l'on construit les différents arcs et nœuds progressivement. Pour chaque nouveau marquage, on détermine l'ensemble des transitions franchissables et pour chaque transition de cet ensemble, on ajoute un arc vers le nouveau marquage. Si un marquage est déjà présent dans le graphe, on trace simplement l'arc.

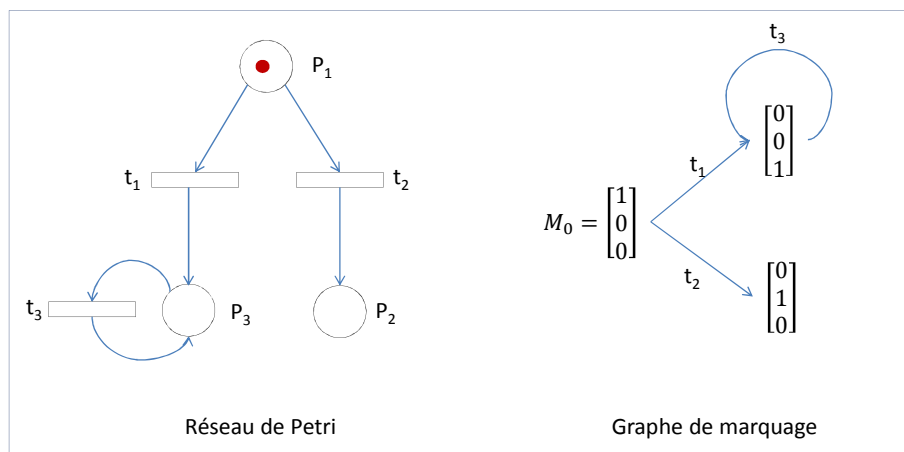


Figure 15 Graphe des marquages

2.4.1.2 Analyse des propriétés

Si nous avons pu extraire un graphe des marquages pour un réseau de Petri, nous pouvons en déduire un ensemble de propriétés :

- **Réseau borné**

Si toutes les places du réseau de Petri marqué sont k -bornées, le nombre de marquages différents est borné et le graphe contient un nombre fini de sommets différents.

On peut en déduire que si le graphe des marquages comporte un nombre fini de sommets, le réseau marqué est **borné**.

- **Régime permanent**

Un réseau de Petri marqué admet une séquence infinie si et seulement si le graphe des marquages possède un circuit.

- **Blocage**

Un réseau de Petri marqué admet un blocage si et seulement si le graphe des marquages contient un sommet sans successeur, appelé marquage puits ou mort.

- **Quasi-vivacité**

Un réseau de Petri marqué est quasi-vivant si et seulement si, pour toute transition t , il existe dans le graphe des marquages un arc d'étiquette t .

- **Vivacité**

Un réseau de Petri marqué est vivant si et seulement si pour toute transition t , toute composante fortement connexe terminale du graphe des marquages contient un arc d'étiquette t . Cette propriété n'est valable que pour les graphes des marquages finis.

La figure suivante présente un réseau de Petri ayant l'ensemble des marquages accessibles fini. On peut donc en extraire son graphe des marquages

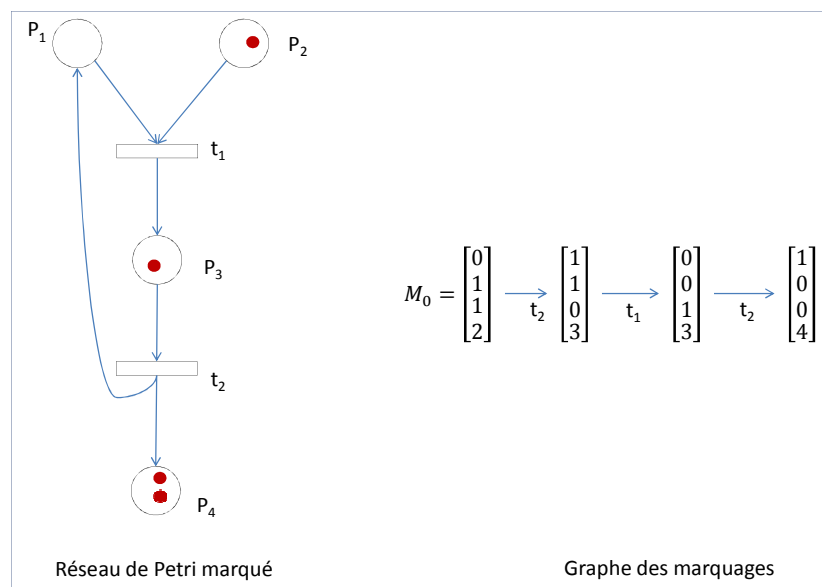


Figure 16 Exemple d'analyse

Avec son graphe nous pouvons constater que le réseau de Petri marqué est : 4-borné, pas vivant, mais quasi-vivant avec un blocage.

2.4.2 Graphe de couverture et analyse des propriétés

Comme cité dans l'introduction, le graphe de couverture est construit à partir d'un réseau qui contient un ensemble d'accessibilité **infini**. Construire un graphe des marquages avec ce réseau est impossible, car il serait lui aussi infini. Dans ce cas, on remplace le graphe des marquages par un graphe de couverture.

2.4.2.1 Graphe

Le graphe de couverture est obtenu à partir de l'arborescence de couverture en identifiant les nœuds dont les étiquettes sont identiques.

Définition : L'arborescence de couverture d'un réseau marqué (N, M_0) est définie par les conditions suivantes :

- La racine r est étiquetée par M_0
- Soit un nœud s d'étiquette \bar{M}
 - S'il existe sur la branche de r à s un nœud de même étiquette \bar{M} , alors s n'a pas de fils
 - Si $\forall t \in T$, on n'a pas $\text{Pré}(t) = W(\blacksquare, t) \leq \bar{M}$, alors s n'a pas de fils
 - Sinon, pour tout t de T tel que $W(\blacksquare, t) \leq \bar{M}$, il existe un nœud s'étiquetant par \bar{M}' , fils de s , le lien entre s et s'étant étiqueté par t . \bar{M}' est défini par :
 - S'il existe sur la branche allant de r à s (r et s compris) un nœud étiqueté par \bar{M}'' tel que $\bar{M}'' < \bar{M} - W(\blacksquare, t) + W(t, \blacksquare)$ alors, pour tout $p \in P$ tel que $\bar{M}''(p) < \bar{M}(p) - W(p, t) + W(t, p)$, on a $\bar{M}'(p) = \S$ et pour les autres places p , $\bar{M}'(p) = \bar{M}(p) - W(p, t) + W(t, p) = \bar{M}''(p)$.
 - S'il n'existe pas de tel sommet, $\bar{M}' = \bar{M} - W(\blacksquare, t) + W(t, \blacksquare)$.

Le symbole \S représente une quantité a priori quelconque de marques. Ce symbole a les propriétés de ∞ :

$$\forall n \in \mathbb{N} \begin{cases} \S + n = \S = n + \S \\ \S - n = \S \\ n < \S \\ \S \leq \S \end{cases}$$

La construction est la même que le graphe des marquages. La différence est que, pour chaque nouveau marquage ajouté, on vérifie s'il n'est pas supérieur à un marquage déjà présent dans au moins une séquence entre M_0 et le nouveau marquage. Si c'est le cas, tous les marquages de place supérieurs sont remplacés par \S . Ce symbole indique que la place en question peut contenir autant de jetons que l'on souhaite (place non bornée). Cette méthode nous donne le graphe de couverture qui est un graphe fini.

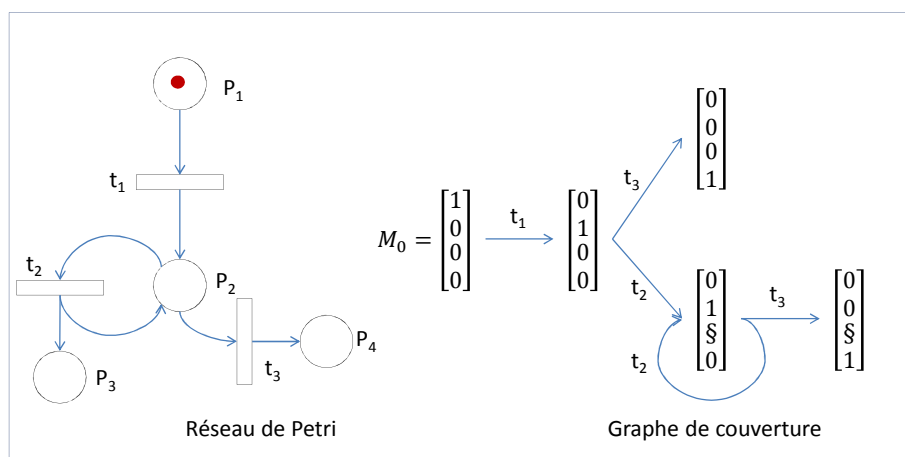


Figure 17 Illustration d'un graphe de couverture

2.4.2.2 Analyse des propriétés

Le passage au graphe de couverture, même s'il permet d'avoir un graphe fini, s'accompagne d'une perte d'information. On pourra exprimer le fait que le contenu d'une place peut grandir

de manière arbitraire, mais on ne saura plus exprimer le fait que cette place pourrait éventuellement être vide. Nous pouvons en déduire un ensemble de propriétés :

- **Réseau non borné**

Lorsque le réseau n'est pas borné, on ne peut pas construire son graphe des marquages contrairement au graphe de couverture.

Le réseau de Petri marqué est borné si le graphe de couverture ne contient pas de nœud avec une composante à la valeur \S . Dans ce cas, le graphe de couverture est identique au graphe des marquages.

- **Régime permanent**

Un réseau de Petri marqué admet un régime permanent si et seulement si le graphe de couverture possède un circuit.

- **Blocage**

Un réseau de Petri marqué admet un blocage si et seulement si son graphe de couverture contient un nœud sans successeur.

- **Quasi-vivacité**

Un réseau de Petri marqué est quasi-vivant si et seulement si toutes les transitions apparaissent dans le graphe de couverture.

- **Vivacité**

Un réseau de Petri marqué est non-vivant si son graphe de couverture contient une composante fortement connexe terminale ne comportant pas toutes les transitions. Si le graphe de couverture est fortement connexe ou si toutes les composantes fortement connexes terminales contiennent chacune toutes les transitions, on peut seulement subodorer que le graphe est vivant mais il est impossible de le démontrer en toute généralité.

Une composante fortement connexe d'un graphe est un sous-graphe tel qu'il existe un chemin (orienté) entre tout sommet A et tout sommet B de ce sous-graphe.

Le réseau de Petri de la figure 17 est : non-borné, non-vivant, quasi-vivant avec un blocage

2.4.3 Limitation

L'analyse des propriétés des réseaux de Petri avec les graphes se fait principalement à partir d'un marquage initial. Cela implique que si le marquage initial se modifie, on doit refaire le graphe des marquages et de couverture. Ce mode d'analyse peut donc devenir assez pénible et long. De plus, l'analyse à partir des graphes de couverture présente certaines limites :

- Il n'est pas possible de vérifier la propriété de vivacité à partir du graphe de couverture, car celui-ci n'est qu'une approximation du graphe des marquages. Il y a donc une perte d'information. Il est cependant possible de vérifier la non-vivacité.
- L'introduction du symbole \S nous fait perdre de l'information : si une place contient \S , on peut en déduire que la place couvre toutes les marques possibles or, elles ne sont pas toutes accessibles.

2.5 Analyse des propriétés des réseaux par l'algèbre linéaire

Comme dit précédemment, l'analyse par les graphes des marquages ou de couverture dépend d'un marquage initial donné. Les propriétés prouvées sont donc valides uniquement pour ce marquage. L'objectif de cette section est d'étudier les propriétés structurelles du réseau, c'est-à-dire celles qui sont indépendantes du marquage initial. Pour cela, on utilise l'algèbre linéaire.

2.5.1 Formalisation et définition de la représentation des réseaux par matrices

2.5.1.1 Matrice d'incidence

Cette section rappelle et définit formellement les différentes notions citées dans le chapitre « Concepts et notions préliminaires » à la section « Représentation matricielle ». La représentation matricielle des réseaux se fait par la représentation des graphes par matrices d'adjacences. Les matrices ont une ligne par place et une colonne par transition.

Définition : Soit $N = (P = \{p_1, \dots, p_n\}, T = \{t_1, \dots, t_m\}, W)$. Le réseau N est représenté à l'aide de trois matrices :

- La matrice $Pré$, appelée **matrice d'incidence avant** : modélise les préconditions de tir des différentes transitions.

$$Pré = (pré_{i,j})_{i \in 1 \dots n, j \in 1 \dots m, \text{ où } pré_{i,j} = W(p_i, t_j)}$$

- La matrice $Post$, appelée **matrice d'incidence arrière** : modélise les postconditions de tir des différentes transitions.

$$Post = (post_{i,j})_{i \in 1 \dots n, j \in 1 \dots m, \text{ où } post_{i,j} = W(t_j, p_i)}$$

- La matrice C , appelée **matrice d'incidence** : modélise les effets du tir des différentes transitions.

$$C = Post - Pré$$

2.5.1.2 Marquage et évolution

Pour déterminer si une transition est franchissable, on utilise la matrice $Pré$. La transition t_j est franchissable depuis M si et seulement si $Pré(\blacksquare, t_j) \leq M$, c'est-à-dire si la $j^{ème}$ colonne de $Pré$ est inférieure ou égale à M . C'est-à-dire que la transition T_j est franchissable si le nombre de jetons présents dans chaque place est supérieur ou égal à la valuation de l'arc reliant la place à la transition.

Définition : Une transition T_j est franchissable si et seulement si $\forall p_i \in P, M(p_i) \geq Pré(p_i, t)$

On aboutit alors au marquage M' défini par $M' = M - Pré(\blacksquare, t_j) + Post(\blacksquare, t_j)$.

$$M' = M + C \times \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} j$$

Nous avons :

$$Pré = \begin{pmatrix} x & y & z & t \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 2 \end{pmatrix} \begin{matrix} P_1 \\ P_2 \\ P_3 \end{matrix} \quad Post = \begin{pmatrix} x & y & z & t \\ 2 & 0 & 1 & 0 \\ 0 & 0 & 4 & 2 \\ 0 & 3 & 0 & 0 \end{pmatrix} \begin{matrix} P_1 \\ P_2 \\ P_3 \end{matrix} \quad C = \begin{pmatrix} x & y & z & t \\ 2 & -1 & 0 & 0 \\ -1 & -1 & 4 & 1 \\ 0 & 3 & 0 & -2 \end{pmatrix} \begin{matrix} P_1 \\ P_2 \\ P_3 \end{matrix}$$

$$M_0 = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix}$$

Franchissement de la transition y:

$$M_0 = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} \quad \text{et} \quad Pré(y, \blacksquare) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} \leq M_0$$

Donc y est franchissable depuis le marquage initial et le tir y conduit au marquage :

$$M = \begin{pmatrix} 1 \\ 3 \\ 2 \end{pmatrix} + \begin{pmatrix} 2 & -1 & 0 & 0 \\ -1 & -1 & 4 & 1 \\ 0 & 3 & 0 & -2 \end{pmatrix} \times \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ 5 \end{pmatrix}$$

Figure 18 Exemple d'un franchissement de transition

On peut étendre cette équation aux séquences de transitions.

Définition : Le vecteur caractéristique d'une séquence de franchissements σ , noté $\bar{\sigma}$, est le vecteur de \mathbb{N}^m tel que $\bar{\sigma}_j$ est le nombre de franchissements de la transition t_j dans la séquence σ .

Si σ est une séquence franchissable depuis le marquage M , et si $M(\sigma > M'$ et $M \geq Pré(\blacksquare, \sigma)$, on a l'équation fondamentale suivante :

$$M' = M + C \times \bar{\sigma}$$

Nous avons :

$$C = \begin{pmatrix} x & y & z \\ 0 & -1 & 1 \\ 0 & 1 & -1 \\ 1 & 0 & -2 \end{pmatrix} \begin{matrix} P_1 \\ P_2 \\ P_3 \end{matrix} \quad M_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

Séquence de franchissement: $\sigma = x^4 y z y z x^3 y z y$ $\sigma = \begin{pmatrix} 7 \\ 4 \\ 3 \end{pmatrix}$

$$M = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & -1 & 1 \\ 0 & 1 & -1 \\ 1 & 0 & -2 \end{pmatrix} \times \begin{pmatrix} 7 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}$$

Figure 19 Exemple d'une séquence de transitions

Remarque : Les résultats de l'équation fondamentale, même s'ils sont toujours calculables, n'ont de sens que si la séquence est effectivement franchissable.

Maintenant que nous avons tous les outils en main, nous pouvons passer à l'analyse d'un réseau.

2.5.2 Analyse des propriétés structurelles des réseaux

2.5.2.1 Invariant de places

Soit un vecteur f de pondération des places : $f: \forall p: f(p) \geq 0$.

L'accroissement pondéré par f du nombre de marques dans le réseau suite au franchissement de la séquence σ : $fM' - fM_0 = fC \times \bar{\sigma}$

Définition : On appelle **invariant de places** toute forme linéaire du marquage des places dont la valeur est constante et ne dépend pas du marquage initial.

Le vecteur f correspond à un invariant de places si et seulement si $fM = \text{constante} = fM_0$ pour tout marquage accessible M , ce qui signifie que pour toute séquence franchissable σ , on a $fC \times \bar{\sigma} = 0$. On en déduit que $fC = 0$.

Propriété : le réseau de Petri est structurellement borné si et seulement si $\exists f > 0, fC \leq 0$

Propriété : le réseau de Petri est structurellement non-vivant si et seulement si $\exists f \geq 0, fC < 0$

Définition : On appelle **semi-flots de place** (*P-semi-flot*), les vecteurs f tels que $fC = 0$ et $\forall i f_i \in \mathbb{N}$

Définition : On appelle **flots de place** (*P-flot*), les vecteurs f tels que $fC = 0$ et $\forall i f_i \in \mathbb{Z}$

L'existence des semi-flots de places permet de vérifier la bornitude d'un réseau : si la matrice C n'admet que des semi-flots de places, le réseau est borné.

2.5.2.2 Invariant de transitions

On a $T(\sigma)$: ensemble des transitions qui apparaissent dans la séquence de franchissements σ .

Définition : Un ensemble D de transitions est une **composante répétitive stationnaire** si et seulement s'il existe une séquence de franchissements σ telle que $T(\sigma) = D$ et $C \times \sigma = 0$.
 D est une **composante répétitive croissante** si $C \times \sigma > 0$

Définition : On appelle **T-semi-flot**, tout vecteur σ tel que $C \times \sigma = 0$

Lorsque l'on a trouvé un T-semi-flot, il reste à prouver qu'il existe bien une séquence effectivement franchissable. Il faut donc trouver un ordonnancement correct des transitions qui compose la séquence et prouve que celle-ci est valide.

2.6 Méthode de réduction

Lors de l'analyse de réseau de Petri avec les graphes des marquages ou de couverture, on peut être confronté à certains problèmes : le graphe construit peut s'avérer conséquent et donc difficile à construire. L'objectif de cette section est de donner un ensemble de techniques de réduction permettant de diminuer la taille du réseau à analyser tout en conservant ses propriétés essentielles. Ces modifications réduiront considérablement la taille du graphe associé. Les opérations de réduction consistent à supprimer ou fusionner des places et des transitions. Ces opérations préservent les propriétés du réseau initial telles que la vivacité, la quasi-vivacité, le caractère borné, l'existence d'un état d'accueil, l'absence de blocage. Ces

propriétés sont vraies dans le réseau réduit si et seulement si elles le sont dans le réseau initial. Il est important de souligner que le réseau simplifié n'est pas équivalent au réseau initial, il ne faut donc pas chercher à l'interpréter. Son rôle est simplement de mettre en évidence les propriétés. Le réseau simplifié ne doit donc pas être interprété dans le monde réel. Il existe 4 réductions principales :

- R_1 : place substituable
- R_2 : place implicite
- R_3 : transition neutre
- R_4 : transitions identiques

2.6.1 Réduction R_1 : Place substituable

Une place est substituable si le franchissement de ses transitions d'entrée est une condition suffisante pour franchir ses transitions de sortie.

Condition: Une place peut être substituable si elle remplit l'une des 3 conditions suivantes :

1. Les transitions de sortie de P_i n'ont pas d'autres places d'entrée que P_i
2. Il n'existe pas de transition T_j qui soit à la fois entrée et sortie de P_i
3. Au moins une transition de sortie de P_i n'est pas une transition puits

Méthode : Supprimer la place. Ensuite, supprimer les transitions en entrée et sortie de la place et créer une transition $t_{i,j}$ par couple (t_i, t_j) des transitions en entrée et sortie dont les places en entrée sont celle de t_i et les places en sortie celle de t_j .

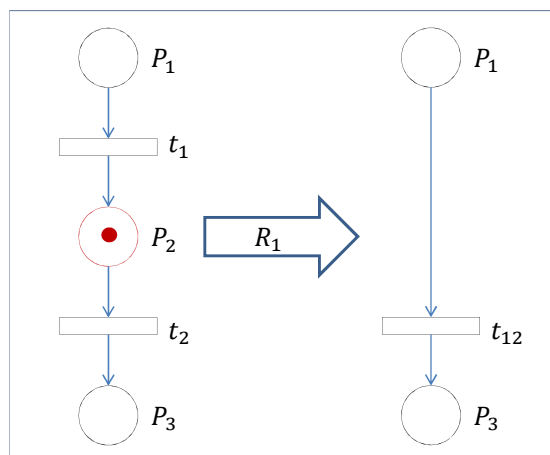


Figure 20 Exemple 1 de réduction R_1

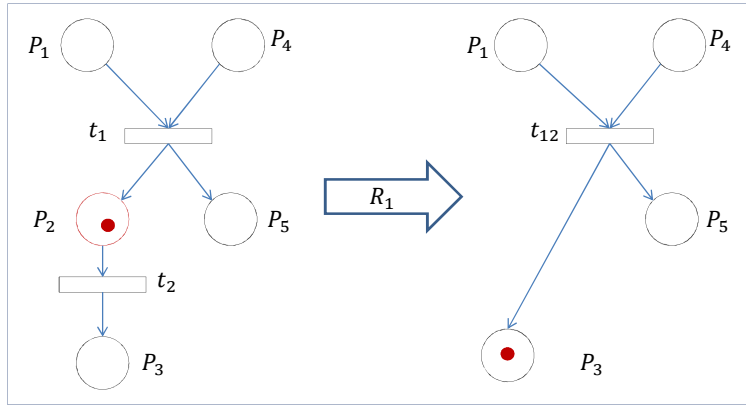


Figure 21 Exemple 2 de réduction R_1

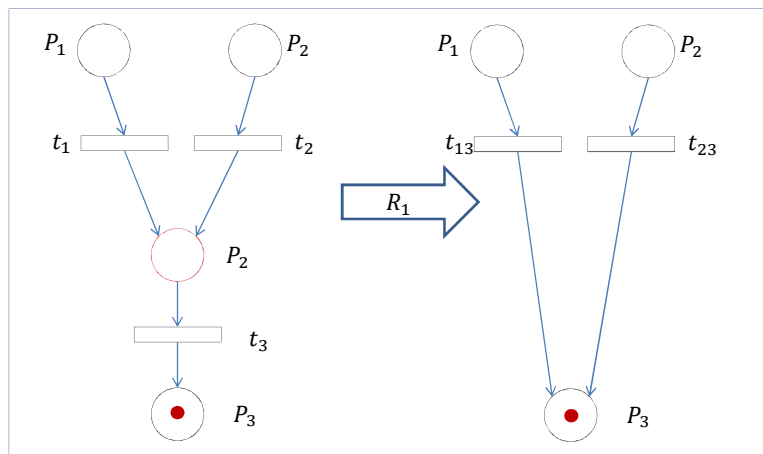


Figure 22 Exemple 3 de réduction R_1

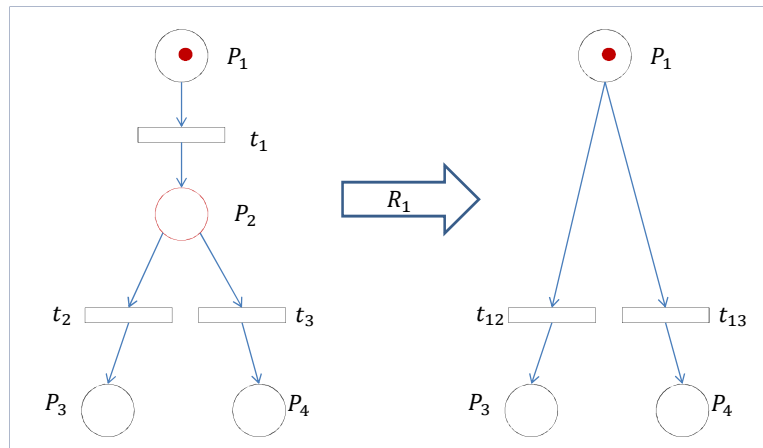


Figure 23 Exemple 4 de réduction R_1

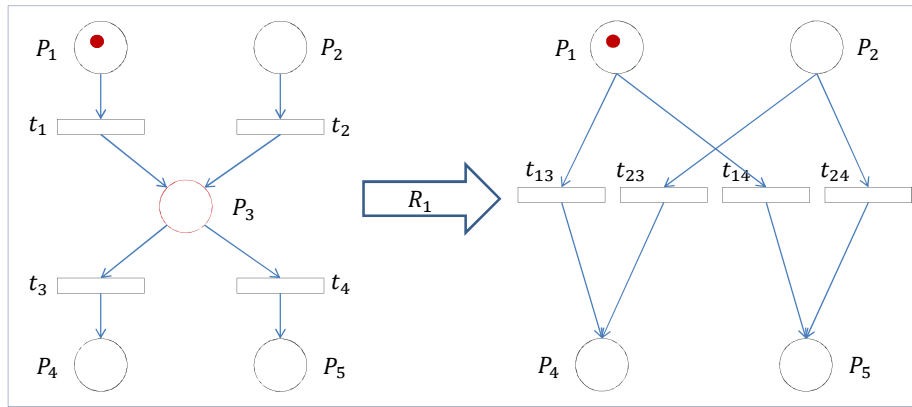


Figure 24 Exemple 5 de réduction R_1

Si la place est marquée, deux situations sont possibles :

1. Il y a une seule transition de sortie, on place la marque dans la place suivante.

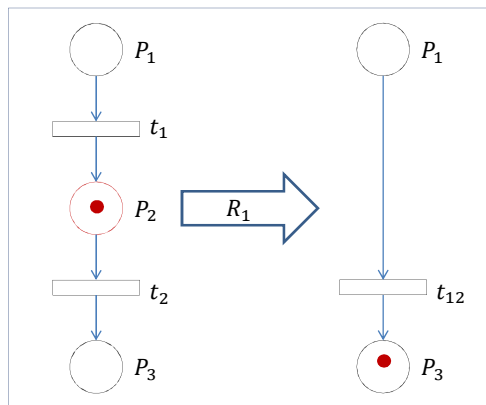


Figure 25 Réduction R_1 avec marquage: une seule transition

2. Il y a plusieurs transitions de sortie, on doit considérer tous les cas possibles. On obtient plusieurs réseaux réduits.

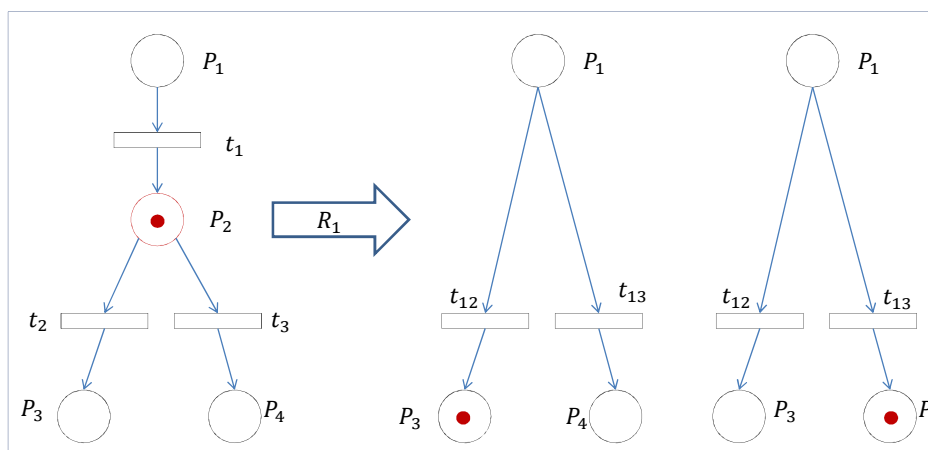


Figure 26 Réduction R_1 avec marquage: plusieurs transitions

2.6.2 Réduction R_2 : Place implicite

Définition : Une place (P_i) est implicite si

1. Son marquage n'a aucun impact sur le franchissement de ses transitions en sortie
2. Son marquage est une combinaison linéaire des marquages des autres places du réseau.

La combinaison linéaire est la suivante :

$$M(P_i) = (\sum_{k \neq i} a_k \cdot M(P_k)) + b \text{ où } a \text{ et } b \text{ sont des nombres rationnels positifs.}$$

Méthode : Supprimer la place et les arcs associés

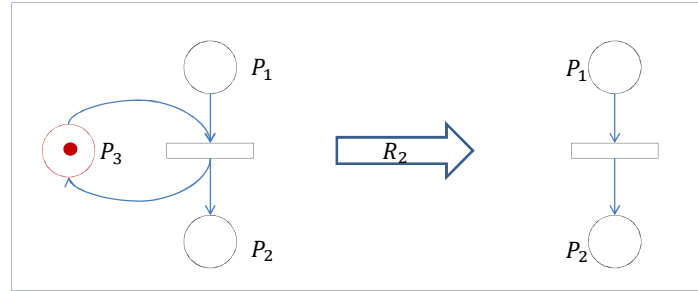


Figure 27 Réduction R_2 : place implicite

Remarque : la réduction de la figure 27 est correcte car la place P_3 est marquée.

Dans la figure suivante, le réseau (a) peut être réduit car les deux conditions sont remplies. On peut supprimer P_2 . Le réseau (b) ne peut être réduit car la condition 1 est remplie mais pas la condition 2. Le réseau (c) ne peut lui non plus être réduit car la condition 1 n'est pas remplie alors que la condition 2 l'est.

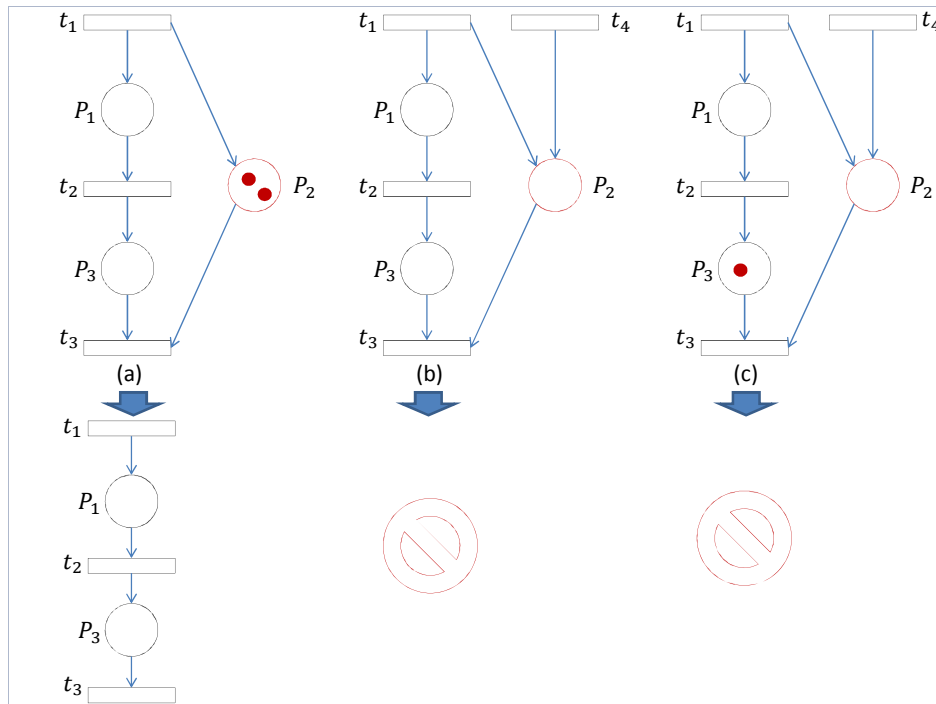


Figure 28 Réduction R_2 : vérification des conditions

2.6.3 Réduction R_3 : Transition neutre

Définition : Une transition est neutre si et seulement si l'ensemble de ses places d'entrée est identique à celui de ses sorties.

Condition : On peut supprimer une transition neutre T_j avec ses arcs entrant et sortant si et seulement s'il existe une transition $T_k \neq T_j$ tel que :

$$Post(P_i, T_k) \geq Pré(P_i, T_j) \text{ pour toute place en entrée de } T_j$$

Méthode : Supprimer la transition et les arcs en entrée et sortie.

Dans la figure suivante, $Post(P_1, t_1) = Pré(P_1, t_5)$ et $Post(P_2, t_1) = Pré(P_2, t_5)$

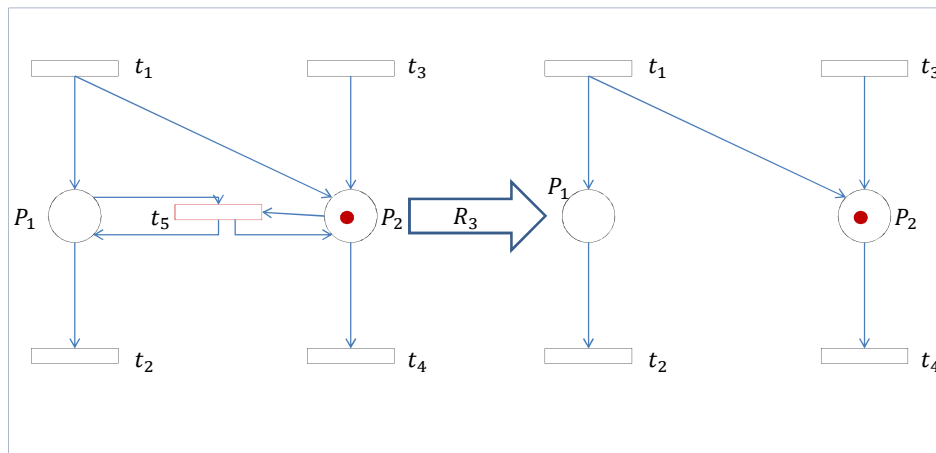


Figure 29 Réduction R_3 : transition neutre

2.6.4 Réduction R_4 : Transitions identiques

Définition : Deux transitions T_j et T_i sont identiques si elles ont le même ensemble de places d'entrée et le même ensemble de places de sortie.

Méthode : Supprimer l'une des deux transitions ainsi que les arcs associés.

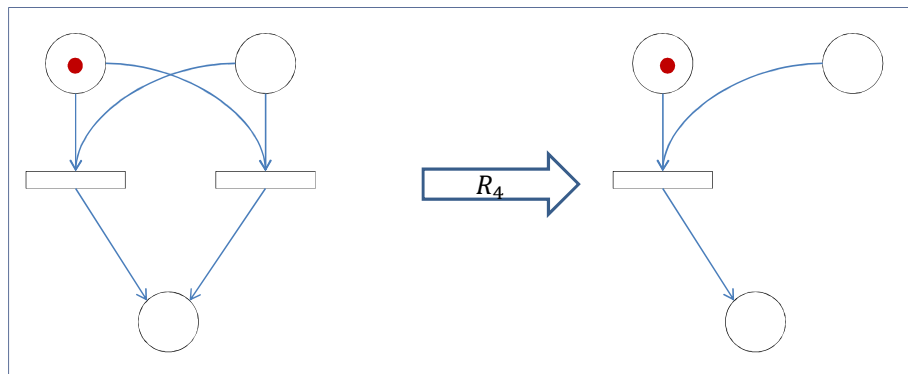


Figure 30 Réduction R_4 : transitions identiques

La figure suivante est un exemple de réduction d'un réseau. Elle contient le réseau de Petri et son réseau réduit.

On supprime dans un premier temps la place P_6 et la transition t_5 par la règle de réduction R_1 .

On supprime ensuite la transition P_8 et la transition T_8 par la règle de réduction R_1 .

Troisièmement, On supprime la place P_9 par la règle de réduction R_2 .

On peut maintenant supprimer par la règle de réduction R_1 la place P_7 et la transition t_6 .

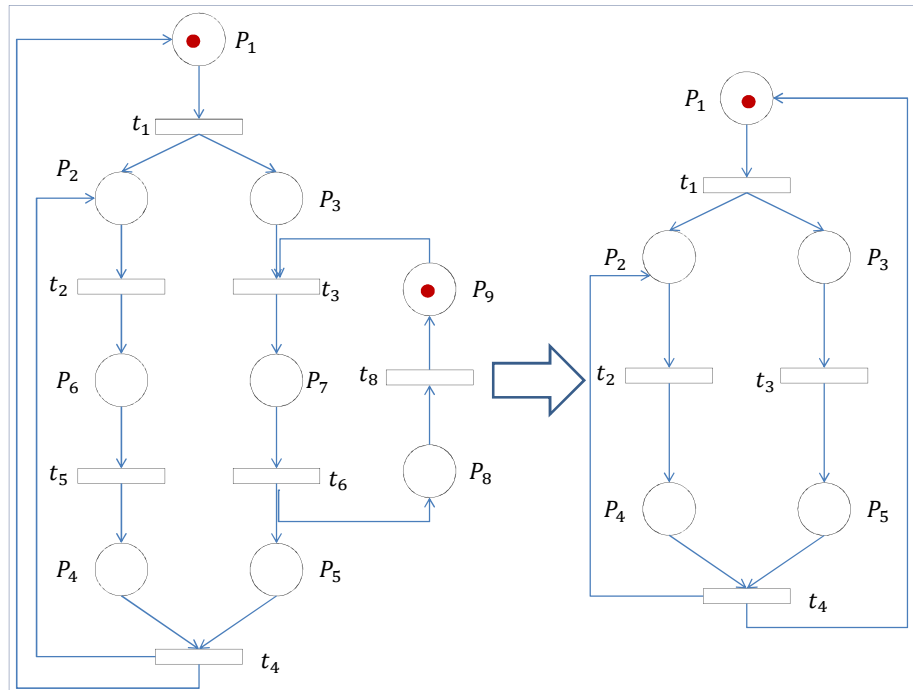


Figure 31 Réduction totale d'un réseau de Petri

CHAPITRE 3 : Quelques extensions des réseaux de Petri

Ce chapitre définit la sélection d'extensions des réseaux de Petri qui sera étudiée dans ce mémoire. Cette sélection est illustrée à la figure 6 du chapitre 1. Certaines extensions, qui ont fait l'objet d'études et de thèses plus approfondies, bénéficieront d'un chapitre à part entière.

3.1 Réseau à arc inhibiteur (Inhibitor arc)

3.1.1 Propriété manquante aux réseaux de Petri ordinaires

Dans les réseaux de Petri ordinaires, les réseaux places/transitions ne permettent pas de modéliser le test à zéro, c'est-à-dire de pouvoir franchir une transition lorsque la place en entrée ne contient aucun jeton. Cette propriété est inexistante à cause de la propriété de monotonie.

3.1.2 Définition

Pour pouvoir tirer une transition en l'absence de jetons, on ajoute au réseau de Petri un nouveau type d'arc : l'arc inhibiteur. Celui-ci relie une place vers une transition. Il faut noter que le tir d'une transition avec un arc inhibiteur ne modifie pas le marquage de la place en amont.

Définition : On appelle réseau de Petri à arc inhibiteur un réseau défini par $N = (P, R, W)$ où P et T sont des ensembles finis disjoints respectivement de places et de transitions et $W : P \times T \cup T \times P \rightarrow \mathbb{N} \cup \{\perp\}$

La règle de franchissement devient alors :

Définition : Soit N un réseau à arc inhibiteur, soit M un marquage et t une transition, t est franchissable depuis M si et seulement si $\forall p \in P$ on a :

1. Si $W(p, t) \in \mathbb{N}$ alors $M(p) \geq W(p, t)$,
2. Si $W(p, t) = \perp$ alors $M(p) = 0$.

On considère le réseau de Petri à arc inhibiteur illustré ci-dessous :

1.

2.

La

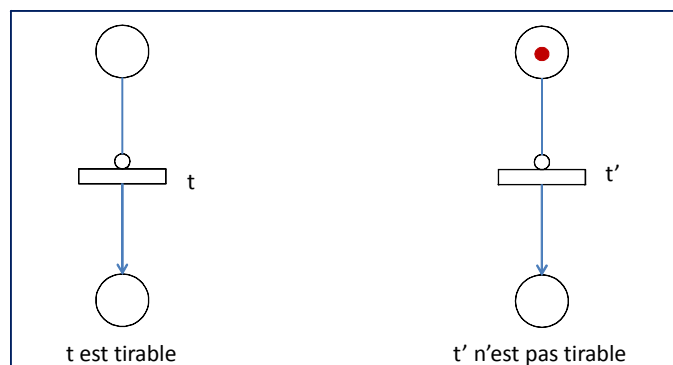


Figure 32 L'arc inhibiteur et transition tirable

3.1.3 Caractéristiques

Grâce aux arcs inhibiteurs, la puissance d'expression des réseaux de Petri ordinaires est améliorée, car ceux permettent de représenter des systèmes nécessitant un test à zéro. Mais cette extension réduit les possibilités d'analyse des réseaux, car il n'est plus possible de vérifier les propriétés du réseau telles que l'accessibilité, la vivacité et le bornage. De plus, les réseaux à arc inhibiteur ne peuvent être représentés dans une matrice de connectivité. Donc, nous perdons aussi toutes preuves mathématiques des réseaux.

Il est cependant possible, dans certains cas, de modéliser le test à zéro sans utiliser les arcs inhibiteurs. Cette alternative est possible en ajoutant des **places complémentaires** au réseau. Pour tester qu'une place est vide sans arc inhibiteur, il suffit de remplacer les arcs par des places complémentaires. Cependant, cet ajout ne peut être réalisé uniquement si la place est initialement bornée.

Théorème : *S'il y a un arc inhibiteur de p vers t et si p est bornée, alors on peut transformer l'arc inhibiteur en un réseau places/transitions.*

Une place P_1 est complémentaire de P_2 si et seulement si : $M(P_1) + M(P_2) = M_0(P_1) + M_0(P_2) = M_2$ quel que soit le marquage atteint.

On a donc : $M(P_1) = M_2 - M(P_2)$. Par conséquent : $M(P_2) = 0 \Leftrightarrow M(P_1) = M_2$

Pour construire le complémentaire de P_2 qui est borné à k , on ajoute une place P_1 de telle sorte que $M(P_1) + M(P_2)$ soit un invariant égal à k . Pour cela, on rajoute des arcs de telle sorte que :

- Les transitions t en sortie de P_2 sont les transitions en entrée de P_1
- $Post(P_2, t) = x \rightarrow Pré(t, P_1) = x$
- Les transitions s en entrée de P_2 sont les transitions en sortie de P_1
- $Pré(P_2, s) = x \rightarrow Post(s, P_1) = x$

L'arc inhibiteur de P_2 vers t est remplacé par un arc de P_1 vers t et par un arc de t vers P_2 évalué à k . La figure suivante illustre cette transformation.

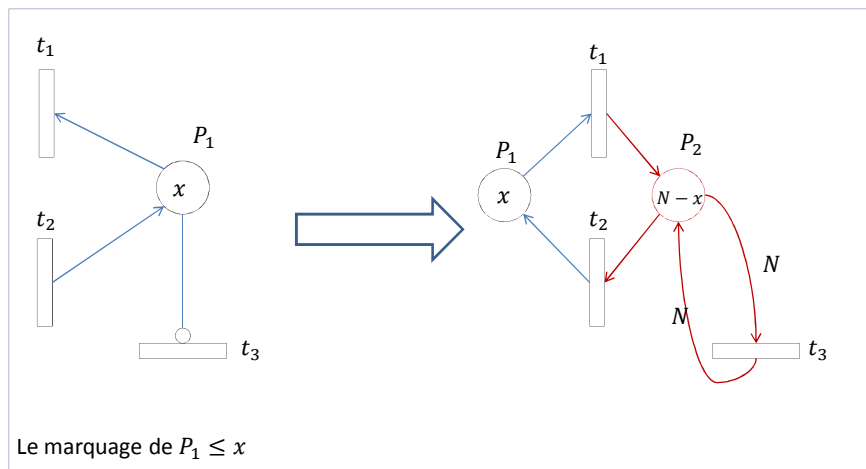


Figure 33 Transformation d'un arc inhibiteur en place complémentaire

3.2 Réseau synchronisé (Synchronized Petri Nets)

3.2.1 Propriété manquante aux réseaux de Petri ordinaires

Dans les réseaux de Petri ordinaires, les réseaux places/transitions permettent de tirer une transition lorsque toutes les conditions sont réunies. Cependant, ces réseaux n'ont aucune interaction avec le contexte d'exécution du système qu'il modélise. Or un ensemble d'actions, d'évènements extérieurs, pourrait ou devrait influencer le franchissement d'une transition.

Par exemple, dans un distributeur bancaire, le système doit pouvoir se lancer lors de l'insertion d'une carte dans l'appareil. L'insertion est donc une action du contexte extérieur qui permet au système d'évoluer.

Cette nouvelle extension permet principalement de modéliser des systèmes contrôle/commande ou des systèmes utilisant des capteurs.

3.2.2 Définition

Un réseau de Petri synchronisé est un réseau de Petri auquel on associe un évènement du contexte d'exécution à chaque transition.

Un évènement est une action du contexte extérieur qui est caractérisé par l'instant auquel il se produit.

Définition : Un réseau de Petri synchronisé est un triplet $(N, E, Sync)$ où :

- N est un réseau places/transitions
- E est un ensemble fini d'évènements externes
- $Sync : T \rightarrow E \cup \{e\}$ est la fonction de synchronisation.
 - Où T est l'ensemble des transitions de N
 - e est l'évènement toujours occurrent (évènement qui se produit en permanence)

Le réseau de Petri est totalement synchronisé si aucune transition n'est associée à e .

Avec l'ajout d'évènement au réseau de Petri, la règle de franchissabilité est redéfinie :

Définition : Une transition t est tirable à un instant u depuis un marquage M si et seulement si :

- t est valide à partir de M
- L'évènement $Sync(t)$ se produit à l'instant u .

Dans un réseau de Petri synchronisé, une transition est franchissable lorsqu'elle satisfait la condition de marquage et que l'évènement externe associé se produit. Si un évènement survient et que plusieurs transitions sont valides à partir du marquage, seules les transitions associées à l'évènement seront tirées. Nous faisons l'hypothèse qu'il ne survient qu'un évènement à la fois.

Il est important de noter que si une transition n'est pas validée par son marquage et qu'un évènement associé à celle-ci survient, la transition ne pourra être franchie. L'évènement sera perdu. Il n'y a pas de mémorisation d'évènement.

Illustrons ceci par l'exemple d'un traitement de texte. Le traitement de texte est en mode « Attente » (Figure A). Lorsque l'utilisateur appuie sur une touche du clavier (L'évènement est inscrit entre parenthèses), cela envoie un signal au système qui se met ensuite en mode « Ecriture » (Figure B). Il inscrit la lettre et se remet ensuite en état d'attente (Figure A). Si l'utilisateur appuie sur une lettre durant la phase d'écriture, celle-ci ne sera pas prise en compte, car la transition « Saisie d'une lettre » n'est pas valide par son marquage. Il faudra attendre que la transition « Saisi d'une lettre » soit valide par son marquage et qu'une nouvelle occurrence de « Clavier » se produise.

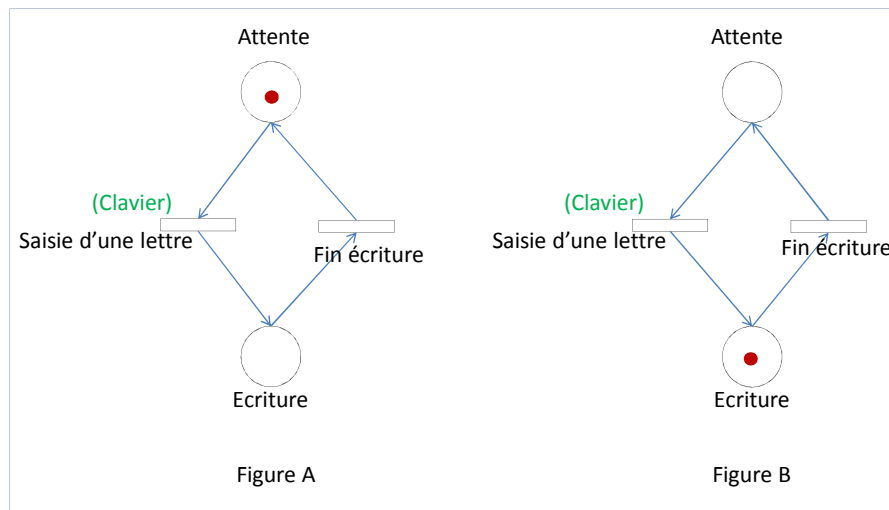


Figure 34 Illustration de Réseau synchronisé : traitement de texte

Dans le cas où deux transitions (t_1 et t_2) qui se suivent et qui sont liées au même évènement (E_1) : la transition t_1 est validée et le franchissement de celle-ci se produit quand l'évènement E_1 survient. Cela pourrait rendre valide la transition, t_2 mais celle-ci n'était pas valide au moment où l'évènement s'est produit donc la transition t_2 ne pourra être franchie. C'est le principe de la **non-réentrance** (voir figure ci-dessous).

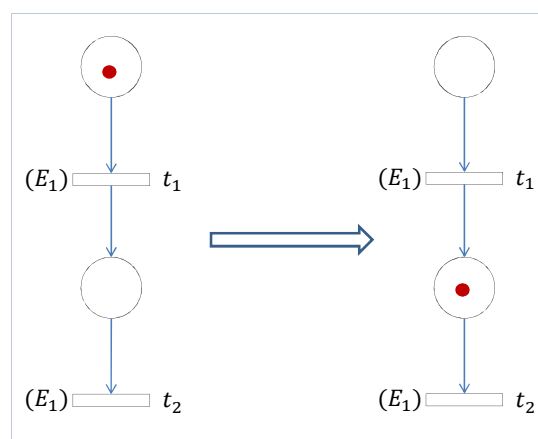


Figure 35 Exemple de deux transitions associées au même évènement

3.2.2.1 Évènement continu e

Cet évènement est toujours occurrence. Une transition associée à cet évènement est tirable dès qu'elle est valide. Cet évènement permet de créer une séquence de transitions et non une unique transition à un évènement.

Le marquage qui valide une ou plusieurs transitions synchronisées sur e est dit **instable**. Cela signifie que le système ne restera pas dans cet état et que le marquage instable a une durée nulle. L'exemple de la figure suivante illustre la notion d'évènement continu.

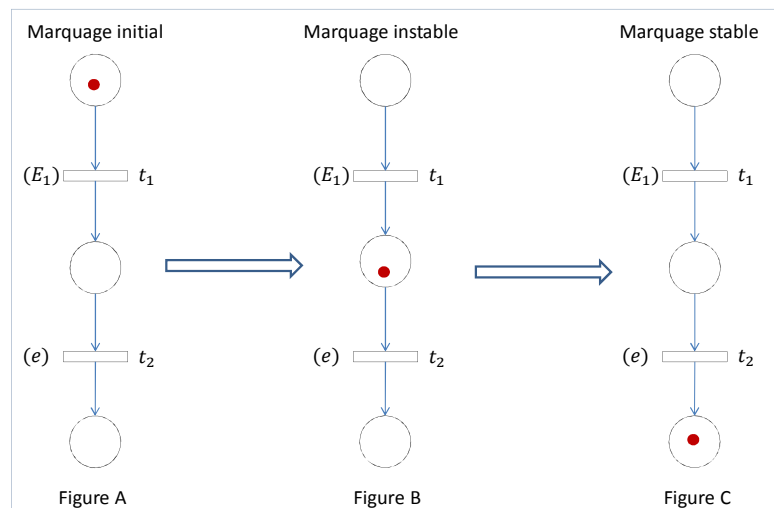


Figure 36 Évènement continu

En démarrant de la figure A et en lançant l'évènement E_1 , on peut tirer la transition t_1 . Le marquage atteint (figure B) se trouve dans un état instable, car celui-ci valide la transition t_2 qui est synchronisée sur e . La transition t_2 est donc directement franchie et nous arrivons de nouveau à un marquage stable (figure C). Finalement, l'occurrence de E_1 a déclenché le tir de la séquence $t_1.t_2$. On parle de **tir itéré** sur l'occurrence E_1 .

3.2.2.2 Deux transitions pour un même évènement

Il peut exister dans le réseau un ensemble de transitions valides qui est associé à un même évènement. Or, les réseaux de Petri ne tolèrent pas le franchissement simultané de plusieurs transitions.

Pour résoudre ce problème, il est nécessaire d'introduire la notion d'ensemble de transitions **persistantes**, c'est-à-dire des transitions valides et qui restent valides après le franchissement non défini d'une d'entre elles.

Considérons un ensemble persistant de transitions valides synchronisées $\{t_1, t_2, \dots, t_k\}$ sur le même évènement. Le déclenchement de cet évènement permet de franchir toutes les permutations possibles de cet ensemble, et le marquage atteint ne dépend pas de l'ordre dans lequel ces transitions ont été franchies. L'ensemble des transitions franchies est une séquence de simulation complète.

Si le réseau contient un évènement e et qu'un événement se produit, on construit d'abord une séquence complète, ensuite on poursuit le tir pour les transitions synchronisées sur e . Il est obligatoire de terminer de construire une séquence complète même sur l'évènement e avant de

regarder les transitions nouvellement validées. Une séquence S se déclare de la manière suivante :

- S est franchissable depuis M et ne comporte que des transitions de $T(X, M)$
- Une transition apparaît au plus une fois dans S
- Toute séquence S' obtenue par permutation de S est aussi franchissable
- S est maximal : il n'existe pas de séquence plus longue

Par exemple, pour le réseau synchronisé de la figure suivante, si l'évènement E_1 se produit, les transitions $\{t_1, t_2\}$ vont être tirées. Les transitions t_4 et t_3 deviennent valides et étant synchronisées sur l'évènement e , elles sont tirées à leur tour. Ce nouveau marquage, instable, valide la transition t_5 .

Le tir itéré sur occurrence de E_1 consiste en la séquence $\{t_1, t_2\} \cdot \{t_4, t_3\} t_5$. (Quand une séquence de simulation complète se limite à une transition, on peut omettre les parenthèses).

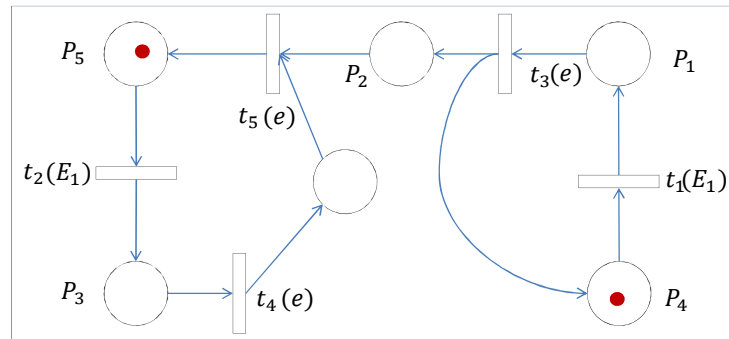


Figure 37 Exemple de tir itéré sur occurrence d'un évènement

3.2.3 Caractéristiques

Avec l'introduction des réseaux de Petri synchronisés, nous pouvons introduire la notion de réseau de Petri **autonome** et **non autonome**.

Un réseau de Petri autonome décrit le fonctionnement d'un système de façon autonome, c'est-à-dire dont l'évolution est régie par ses propres lois. Il n'y a aucune d'interaction avec l'environnement externe. Tous les réseaux de Petri étudiés plus haut sont donc des réseaux autonomes.

Les réseaux de Petri synchronisés sont des réseaux de Petri non autonomes, car ils dépendent et réagissent avec l'environnement externe.

Pour l'analyse des réseaux de Petri synchronisés bornés, il faudra réaliser la construction du graphe des marquages stables. Ce graphe des marquages contient des arcs étiquetés par un couple <séquence itérée sur occurrence d'un évènement/évènement> et des nœuds étiquetés par des marquages accessibles stables. Dans le cas des réseaux non bornés, on construira le graphe de couverture stable.

Propriété : Un réseau de Petri synchronisé est borné si, pour tout marquage stable et instable, il est borné.

Propriété : Une transition est vivante si pour tout marquage accessible, il existe une séquence d'évènements externes telle que cette transition soit franchie.

3.2.3.1 Lien entre un réseau synchronisé et un réseau autonome

Les propriétés de bornitude et de vivacité d'un réseau de Petri autonome ne sont pas toujours conservées dans le réseau de Petri synchronisé associé.

Propriété : Un réseau de Petri synchronisé est borné pour un marquage initial M_0 si le réseau de Petri autonome associé est borné pour le marquage initial M_0 . Mais la réciproque n'est pas forcément vraie.

Il est logique que le réseau de Petri synchronisé soit borné si le réseau de Petri autonome associé est borné, car l'ensemble des marquages accessibles de ce premier est inclus dans l'ensemble des marquages accessibles du réseau de Petri autonome associé. Par contre, si le réseau autonome est non-borné, cela n'implique pas obligatoirement que le réseau synchronisé soit non-borné. La figure suivante, illustre ceci.

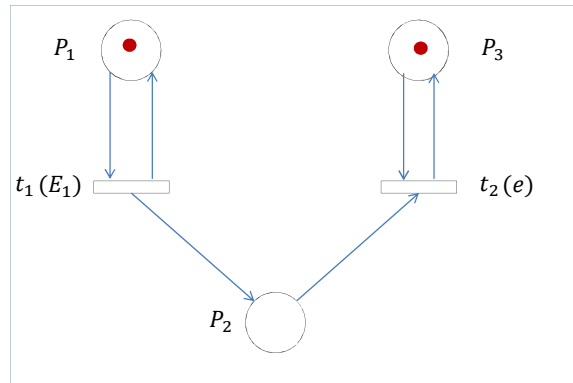


Figure 38 Etude de la bornitude entre réseau autonome et réseau synchronisé

Dans le cas du réseau autonome, on voit que celui-ci n'est pas borné, car il peut avoir autant de jetons dans la place P_2 . Il n'y a pas de restriction entre les jetons qui entrent et qui sortent de cette place. Cette place n'est pas bornée. Cette propriété peut se vérifier par le graphe de couverture qui se trouve sur la figure suivante.

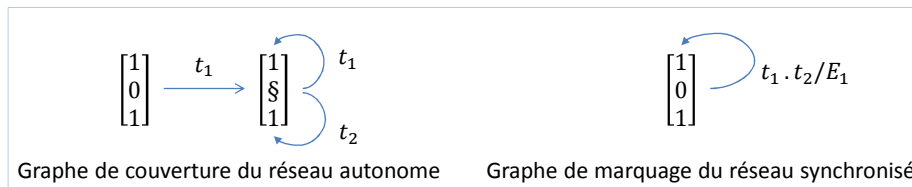


Figure 39 Graphe de marquage et de couverture des réseaux autonome et synchronisé

Dans le cas du réseau synchronisé, l'ajout d'un jeton dans la place se produit lorsqu'une occurrence de E_1 apparaît. Nous nous retrouvons avec un marquage instable, car la transition t_2 devient à son tour valide et peut directement être franchie. La place P_2 est, contrairement au réseau autonome, bornée. Cette propriété peut se vérifier par le graphe des marquages stable qui se trouve sur la figure du dessus.

Propriété : Pour qu'un réseau de Petri synchronisé soit vivant pour un marquage initial M_0 , il est ni nécessaire, ni suffisant que le réseau de Petri autonome associé soit vivant par le marquage initial M_0 .

A la figure suivante, on remarque que le réseau autonome est quasi-vivant car la séquence $t_3 t_2$ est franchissable, ainsi que la séquence $t_1 t_3$. Par contre il n'en est pas de même pour le réseau synchronisé. Seule la séquence franchissable est $t_1 t_3$ sur l'occurrence de E_1 . Le réseau n'est pas quasi-vivant.

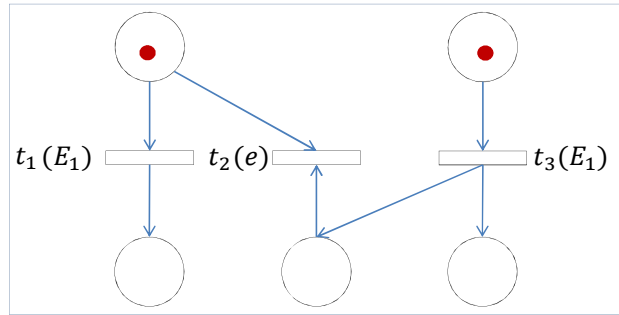


Figure 40 Etude de la vivacité entre réseau autonome et réseau synchronisé

3.3 Réseau de Petri Prédicats/Transitions (Predicate/Transition)

3.3.1 Propriété manquante aux réseaux de Petri ordinaires

Les réseaux de Petri ordinaires ne permettent pas de distinguer les jetons. Or, la modélisation réaliste d'un système nécessite souvent de raisonner aussi sur la nature des jetons, et sur leur transformations. Dans ce but, des extensions associées aux réseaux de Petri permettent l'annotation des réseaux à l'aide d'un langage de premier ordre. Les jetons sont alors des expressions du langage, et les transformations entre un état et un autre sont décrites par les formules annotant les transformations. Le marquage lui-même est un ensemble multiple de jetons, et le tir d'une transition revient à transformer ces jetons. Il existe deux extensions qui permettent de gérer ceci : les réseaux de Petri Prédicats/Transitions et les réseaux colorés. Les réseaux colorés font le sujet d'un chapitre (chapitre 6). Les réseaux Prédicats/Transitions sont basés sur la logique du premier ordre alors que les réseaux colorés sont basés sur un langage fonctionnel.

L'idée de base à l'origine des réseaux Prédicats/Transitions est la possibilité de définir des prédicats dynamiques dont l'interprétation change en fonction des événements et du contexte.

Un exemple permet de saisir cette propriété. Dans un réseau de Petri ordinaire, on peut modéliser ce système : « Si une vis et un écrou faire boulon ». Par contre, il n'est pas possible de modéliser ce système : « Si une vis $\langle b \rangle$ et un écrou $\langle a, g, \rangle$ faire boulon $\langle m \rangle$. Si une vis $\langle l \rangle$ et écrou $\langle c, p \rangle$, faire boulon $\langle n \rangle$ ». On pourra identifier si c'est une vis de type $\langle b \rangle$ ou $\langle l \rangle$ par exemple et on pourra les trier. On doit associer les bons boulons avec les bonnes vis.

Les réseaux de Petri Prédicats/Transition sont utilisés pour modéliser des protocoles de communications, des systèmes de production,...

3.3.2 Définition

Les réseaux de Petri Prédicats/Transitions sont construits de la manière suivante : les prédicats font office de place et les transitions sont remplacées par des règles dans une logique de premier ordre. On associe à un arc une variable.

Définition : Un réseau de Petri Prédicat/Transition est un triplet $PT = \langle R, A_n, M_0 \rangle$ où :

- R est le réseau de Petri ordinaire sous-jacent de PT
- A_n est l'annotation de PT , définie par le 4-uplet $A_n = \langle X, A_x, A_c, A_a \rangle$ où
 - X est un ensemble de variables formelles¹ associé à chaque jeton
 - A_x est une application associant à chaque arc un vecteur de variable de X
 - A_c est une application associant à chaque transition une condition sous la forme d'un prédicat utilisant les variables de X
 - A_a est une application associant à chaque transition une action sous la forme d'une suite d'affectation de valeurs aux variables formelles
- M_0 est le marquage initial du réseau de Petri. Chaque jeton est un vecteur des variables de X . Le marquage initial définit les valeurs des variables du jeton.

¹ Une variable est formelle si elle ne contient aucune valeur. Ici, elles sont sous la forme $\langle x \rangle$.

Dans un réseau de Petri Prédicats/Transitions, une transition est sensibilisée² pour un ensemble défini de jetons de ses places d'entrées. Elle devient sensibilisée si les variables associées à ses arcs d'entrées peuvent être remplacées par les valeurs des variables des jetons de ses places d'entrées.

Elle devient tirable lorsque celle-ci est sensibilisée. Après le franchissement d'une transition, l'action associée à celle-ci redéfinit les valeurs des variables des arcs sortants. De manière plus simple, les variables associées aux jetons ne sont modifiées que par le franchissement de transition.

Illustrons cette extension par l'exemple cité dans l'introduction. La place P_1 modélise le stock de vis et la place P_2 modélise le stock d'écrous. La place P_3 modélise le stock de boulons qui sont soit montés par la transition t_1 . On ajoute à la transition t_1 les conditions suivantes : $x < y < z$

Le réseau (a) peut être tiré car si on associe $x=a$, $y=b$ et $z=g$, la transition est tirable car $a < b < g$. Par contre, dans le réseau (b), la transition n'est pas tirable car si on associe $x=a$, $y=l$ et $z=g$, la condition n'est pas respectée : $x < l < g$ n'est pas valide.

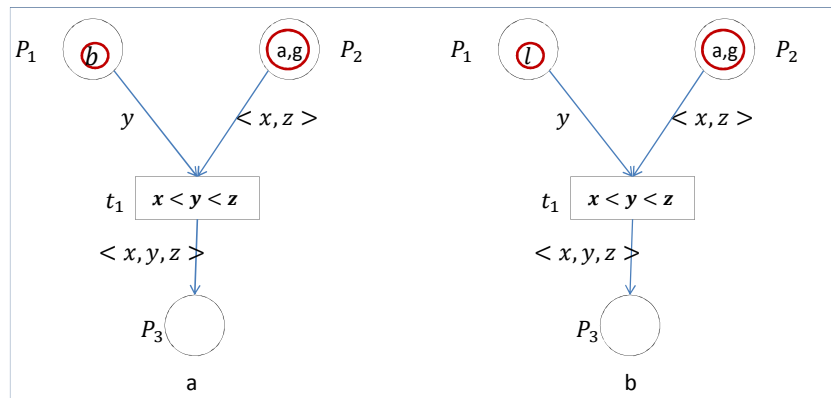


Figure 41 Réseau de Petri Prédicats/Transitions

Lors du franchissement de la transition t_1 , on prend la constante qui représente une vis dans la place P_1 , une autre représentant un écrou dans la place P_2 et on met une nouvelle constante codant le fait que l'on a mis en relation l'écrou avec la vis pour former le boulon dans la place P_3 .

3.3.3 Caractéristiques

Le marquage n'est plus une distribution de constantes dans les places, mais une distribution de n-uplets (liste de constantes). Attention de ne pas confondre les n-uplets $\langle a, b \rangle$, qui sont utilisés dans des franchissements de transition de façon indépendante, avec la somme $\langle a \rangle + \langle b \rangle$ qui devra être substituée à des variables. C'est pourquoi les expressions associées aux arcs sont des sommes de n-uplets de variables formelles.

² Une transition est sensibilisée si et seulement si toute place en entrée contient le nombre de jetons demandé.

3.4 Réseau à files

3.4.1 Propriété manquante aux réseaux de Petri ordinaires

Les réseaux de Petri ordinaires n'ont pas de mémoire, c'est-à-dire qu'ils n'ont pas la notion d'historique et ne peuvent pas modéliser de manière explicite l'ordonnancement d'actions. Cela veut dire que les réseaux ordinaires permettent l'apparition d'une situation de famine³. Or certains systèmes ne peuvent admettre cette situation. Cela s'explique par le fait qu'un réseau ordinaire décrit plus de comportements alors qu'il ne devrait apparaître, dans certains cas, qu'un seul comportement effectif. L'exemple concret de ce type de problème est la gestion de l'accès à une section critique par des sémaphores⁴.

Les sémaphores utilisent des files pour gérer l'accès à la section critique afin d'éviter le risque de famine. On ramène, dans les réseaux ordinaires, le concept de file d'attente du sémaphore. La modélisation de file d'attente permet de mémoriser l'historique des demandes faites.

3.4.2 Définition

Les réseaux à files permettent de gérer des files d'attente. Celles-ci sont modélisées par une file F qui retient les identités des jetons et l'ordre dans lequel ceux-ci sont arrivés. Les réseaux à files sont assez semblables aux réseaux ordinaires. Ils contiennent cependant deux différences : la coloration des jetons qui permettent de les identifier (voir chapitre 6 : les réseaux colorés), ainsi que la modification des opérations de franchissabilité et du tir de transition. Les modifications sont les suivantes :

Réseau de Petri ordinaire	Réseau à files
Entier	Mot
Place	File
Jeton	Lettres formant des mots
Valuation entière	Valuation par mot
Addition	Concaténation
Soustraction	Suppression d'un préfixe

Définition : Un réseau à files est un quadruplet $R = (A, F, T, W)$ où :

- A est un alphabet
- F est un ensemble fini de files
- T est un ensemble fini de transitions (disjoints de F)
- $W: F \times T \cup T \times F \rightarrow A^*$ est la fonction de valuation.

Un **marquage** est une application $F \rightarrow A^*$

A est un alphabet, c'est-à-dire un ensemble fini de symboles. A^* est l'ensemble des mots écrits sur l'alphabet A , c'est-à-dire l'ensemble des suites finies d'éléments de A .

³ La famine est un terme informatique pour désigner un problème que peut avoir un algorithme d'exclusion mutuelle. L'exclusion mutuelle permet d'éviter que des ressources partagées ne soient utilisées en même temps.

⁴ Un sémaphore est une variable qui permet de gérer l'accès à des ressources partagées.

On définit l'opération de **concaténation** sur A^* par : soit $u = u_1 u_2 \dots u_n$ et soit $v = v_1 v_2 \dots v_p$ (u_i et $v_j \in A$) deux mots sur A de longueurs respectives n et p , alors $u.v$ est le mot écrit sur A de longueur $n + p$ défini par $u.v = u_1 u_2 \dots u_n v_1 v_2 \dots v_p$.

On définit une relation sur A^* , appelée **relation préfixe** ou **relation facteur gauche** par : $u <_g v$ si et seulement si $\exists w \in A^*$ tel que $v = u.w$.

Exemple : on a $abbac <_g abbaccabac$

Illustrons cette nouvelle extension : Dans le réseau suivant, nous avons $F = \{F_1, F_2, F_3\}$, $T = \{t_1, t_2, t_3, t_4, t_5\}$, et par exemple $W(t_2, F_3) = ab$. Le marquage initial est $M_0 = (a, b, -)$. L'explication formelle de la franchissabilité est donnée plus bas, mais nous pouvons déjà tirer de ce réseau cet enchainement. Après le tir de t_1 , nous avons le marquage $M = (-, -, ba)$. Cela permet de ne tirer que t_5 . On obtient le marquage suivant : $M = (-, ba, a)$. On tire la transition t_4 , le marquage $M = (ab, ba, -)$. On peut ensuite tirer t_1 pour avoir le marquage $M = (b, a, ba)$. Après avoir tiré t_2 , on obtient $M = (-, -, baab)$

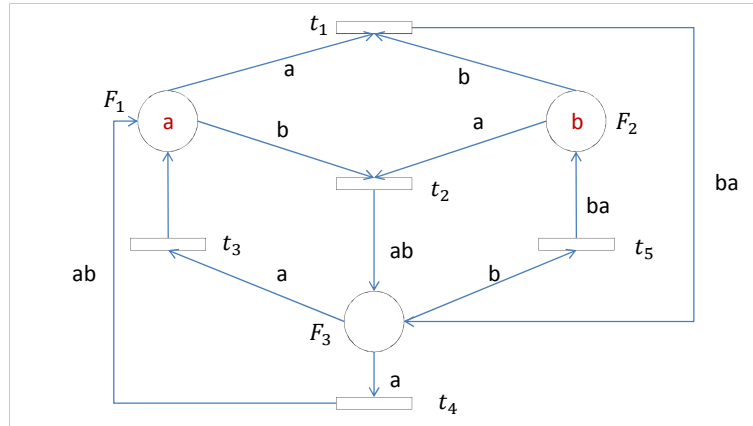


Figure 42 Réseau à files

3.4.3 Caractéristiques

Les réseaux à files ont la même dynamique que les réseaux de Petri ordinaires. Avec l'ajout des nouveaux concepts et modifications dans le réseau de base, nous devons redéfinir les règles de franchissabilité et de tir.

Définition : Une transition est **franchissable** à partir d'un marquage M si et seulement si $\forall f \in F$, on a $W(f, t) <_g M(f)$. On le note toujours $M(t >)$. Lors du **tir** effectif de t , le marquage atteint M' ($M(t > M')$) est défini par la relation : $\forall f \in F, M(f).W(t, f) = W(f, t).M'(f)$.

Pour illustrer cette nouvelle définition, reprenons l'exemple précédent.

$W(F_1, t_1) = a <_g M_0(F_1) = a$ et $W(F_2, t_1) = b <_g M_0(F_2) = b$.

Donc $M_0(t_1 > M)$, avec $M = (-, -, ba)$, puis $W(F_3, t_5) = b <_g M(F_3) = ba$. Donc $M(t_5 > M')$, et $M' = (-, ba, a)$.

Si pour une file f , il existe une lettre a de A telle que $W(f, t), W(t, f)$ et $M_0(f)$ sont tous de la forme a^n avec $n \in \mathbb{N}$, alors f se comporte comme une place de réseau ordinaire. On pourra utiliser la représentation par places.

3.5 Réseau de Petri à objet

3.5.1 Propriété manquante aux réseaux de Petri ordinaires

Les réseaux de Petri ordinaires ont une insuffisance dans le pouvoir d'expression. En effet, les jetons circulant dans un réseau ne contiennent aucune donnée et ne sont donc pas différenciables. Or cette distinction pourrait être souhaitable pour pouvoir lier des structures de données sous la forme de liste d'attributs et des méthodes de transformation de ces données dans un système. Cette nouvelle démarche est contraire à une démarche fonctionnelle suivant laquelle on définit de façon séparée des fonctions et des structures de données que le système devra remplir. A cet effet, une nouvelle notion est rajoutée aux réseaux de Petri : **l'objet**.

Le concept d'objet est de plus en plus présent en Génie Logiciel. Celui-ci permet de construire l'application autour d'entités qui contiennent un ensemble de données et de méthodes manipulant ces données.

3.5.2 Définition

Un réseau de Petri à Objet est un réseau dans lequel on remplace les jetons par des tuples d'objets. Ces nouveaux réseaux peuvent être vue comme une utilisation des réseaux de Petri Prédicats/Transitions, mais, avec une approche objets. On associe au réseau un ensemble de classes d'objet. Les places et les arcs contrôlent et filtrent les types des objets qui circulent dans le réseau.

Définition : Un réseau de Petri à objets est défini par un 9-uplet

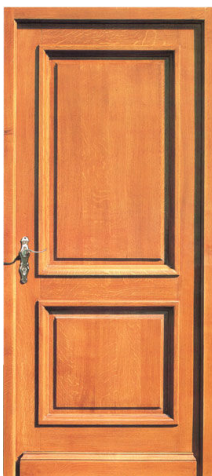
$O = \langle P, T, C_{class}, V, Pr, Post, A_c, A_a, M_0 \rangle$ où :

- P est un ensemble fini de places
- T est un ensemble fini de transitions
- C_{class} est un ensemble fini de classes d'objets, éventuellement organisé en une hiérarchie et définissant pour chaque classe un ensemble d'attributs
- V est un ensemble de variables typées de C_{class}
- Pr est la fonction place précédente qui à chaque arc d'entrée d'une transition fait correspondre une somme formelle de n -uplets d'élément de V
- $Post$ est la fonction place suivante qui à chaque arc de sortie d'une transition fait correspondre une somme formelle de n -uplets d'élément de V
- A_c est une application associant à chaque transition une condition faisant intervenir les variables formelles associées aux arcs d'entrée et les attributs des classes correspondantes
- A_a est une application associant à chaque transition une action faisant intervenir les variables formelles associées aux arcs d'entrée et les attributs des classes correspondantes.
- M_0 est le marquage initial qui associe à chaque place une somme formelle de n -uplets d'instances d'objets (les objets sont représentés par des identificateurs)

3.5.2.1 Les classes

Une classe d'objets est définie par un ensemble d'attributs/propriétés et un ensemble d'opérations/méthodes. Les opérations manipulent les valeurs des attributs. On peut créer une nouvelle classe qui descend d'une autre classe. La nouvelle classe, appelée sous-classe, hérite des attributs et des méthodes de la classe supérieure, dite superclasse. Il est important de retenir que les classes ne sont que des définitions. Un objet est une instance d'une classe d'objets. L'instanciation permet d'affecter des valeurs aux attributs de la classe d'objets et d'exécuter les opérations. L'objet est créé et détruit dynamiquement durant l'exécution du logiciel.

L'exemple suivant illustre un objet « Porte » qui est décrit par des attributs et qui peut effectuer un ensemble d'actions.



ATTRIBUTS	METHODES
- Hauteur	- Ouvrir
- Largeur	- Fermer
- Longueur	- Entre-ouvert
- Épaisseur	- Fermer à clé
- Couleur	
- Texture	

3.5.2.2 Les attributs

En plus des attributs définis dans la classe, un attribut implicite contient le nom de la place où l'objet est localisé. Les opérations associées aux transitions portent sur les attributs des objets situés dans les places d'entrée. L'opération ne pourra être exécutée pour un objet que si celui-ci est localisé dans la place d'entrée. Une notion de contrôle est introduite, car pour que l'opération soit applicable sur les attributs d'une instance d'objet, il faut que celle-ci soit dans un certain état c'est-à-dire qu'il soit dans une certaine place.

3.5.2.3 Illustration

Pour reprendre l'exemple du réseau Prédicats/Transitions, nous avons deux classes, la classe vis et la classe écrou :

$C_{class} = \{vis, écrou, boulon\}$ avec

$$\begin{aligned}
 - vis &= \left\{ \begin{array}{l} nom : \text{identificateur} \\ operations : \text{listes des opérations: viser} \\ date : \text{date de création} \\ type : \text{type de la vis} \end{array} \right. \\
 - écrou &= \left\{ \begin{array}{l} nom : \text{identificateur} \\ operations : \text{listes des opérations} \\ date : \text{date de création} \\ type : \text{type de l'écrou} \end{array} \right.
 \end{aligned}$$

$$- \text{boulon} = \begin{cases} \text{nom} : \text{identificateur} \\ \text{operations} : \text{listes des opérations} \\ \text{date} : \text{date de création} \\ \text{type} : \text{type de boulon} \end{cases}$$

Les conditions et les actions associées aux transitions correspondent aux méthodes et travaillent sur les attributs des objets substitués aux variables formelles lors du franchissement.

La figure suivante représente le réseau de Petri à objet. Pour associer la vis et l'écrou, il faut qu'il soit tous les deux de même type. On associe ensuite la vis à l'écrou en appelant la méthode viser de l'objet vis. Cela va créer un nouvel objet, un boulon.

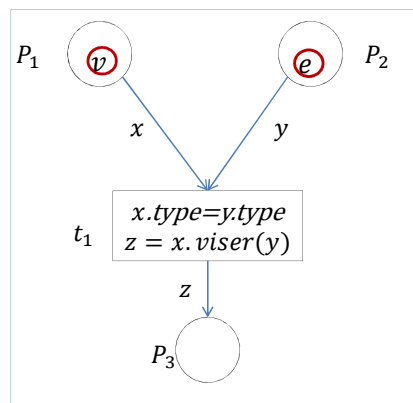


Figure 43 Réseau de Petri à objet

L'ensemble des variables est :

- x de type de vis
- y de type d'écrou
- z de type boulon

Et le marquage initial est définie par :

- v est une classes de vis
- e est une classes d'écrous
- $M_0 = [v, 0, e]$

3.5.3 Caractéristiques

L'étude des réseaux de Petri à objets se fait par l'étude du réseau de Petri sous-jacent (réseau de Petri ordinaire). Grâce aux invariants du réseau de Petri sous-jacent, on peut obtenir des résultats sur la conservation des nombres d'objets indépendamment de leurs classes. Pour obtenir une information plus précise, on procède par projection. Cela consiste à construire un réseau de Petri sous-jacent qui ne contient qu'une classe d'objet. Par exemple, si on souhaite projeter un réseau par rapport à une classe $class_1$ de C_{class} , on remplace dans Pré et Post toutes les variables de classe $class_1$ par 1 et toutes les autres 0. La nouvelle matrice d'incidence contient des entiers naturels. Elle permet de calculer des composantes comme dans un réseau de Petri ordinaire.

3.6 Réseau de Petri flou

3.6.1 Propriété manquante aux réseaux de Petri ordinaires

La logique floue appliquée aux réseaux de Petri permet d'introduire une base de connaissance incertaine du domaine d'application. Cette base de connaissance se modélise par une suite de règles de logique floue. L'utilisation de cette logique permet d'affiner les informations acquises sur le terrain et des interprétations humaines. Cette logique introduit un nouvel élément de type « degré de vérité » (valeur continue comprise entre 0 et 1) qui est associé à chaque affirmation.

Ce nouveau concept peut par exemple s'inscrire dans la réalisation de système de surveillance dynamique, de production, de maintenance et de pronostic en introduisant la possibilité de rester proche de l'opérateur humain. C'est-à-dire en tenant compte de sa façon de raisonner. Ces nouveaux réseaux permettent d'introduire la notion d'incertitude.

3.6.2 Définition

Un réseau de Petri flou est un réseau de Petri dans lequel on peut voir une place comme une condition, une transition comme un lien de causalité de la condition et un jeton dans une place comme un fait pour proclamer la vérité de la condition associée à la place.

Définition : Un réseau de Petri flou est défini par un 8-uplet $RPF = \langle P, T, D, I, O, f, \alpha, \beta \rangle$ où :

- P est un ensemble fini de places
- T est un ensemble fini de transitions
- D est un ensemble de propositions
- I et O sont la fonction d'un ensemble de place d'entrée et de sortie des transitions où :
 - o $I: P \rightarrow T$ est une fonction d'entrée, une correspondance des transitions vers un ensemble de places
 - o $O: T \rightarrow P$ est une fonction de sortie, une correspondance des transitions vers un ensemble de places
- $f: T \rightarrow [0..1]$ est une fonction d'association, une correspondance des transitions vers des valeurs réelles entre 0 et 1
- $\alpha: P \rightarrow [0..1]$ est une fonction d'association, une correspondance des places vers des valeurs réelles entre 0 et 1
- $\beta: P \rightarrow D$ est une fonction d'association, une bijection des places vers les propositions

3.6.2.1 Les grandes lignes de la logique floue

La logique floue est une généralisation de la logique binaire. Elle donne la possibilité de définir des valeurs entre 0 et 1. Par exemple, dans la logique binaire, la température de l'eau est soit chaude soit tiède. Dans la logique floue, la température de l'eau qui est à 35° peut être chaude à 12% et tiède à 65%. Ces pourcentages correspondent à un degré d'appartenance à un ensemble : l'ensemble flou. Par exemple, la température de l'eau peut être perçue de manière différente par chacun d'entre nous. Certain pourrait dire que l'eau est froide entre 0 et 30 degrés et d'autre diront entre 0 et 25 degrés. On définit, dans la figure suivante, trois ensembles flous : froid, tiède et chaud. Ceux-ci peuvent se chevaucher.

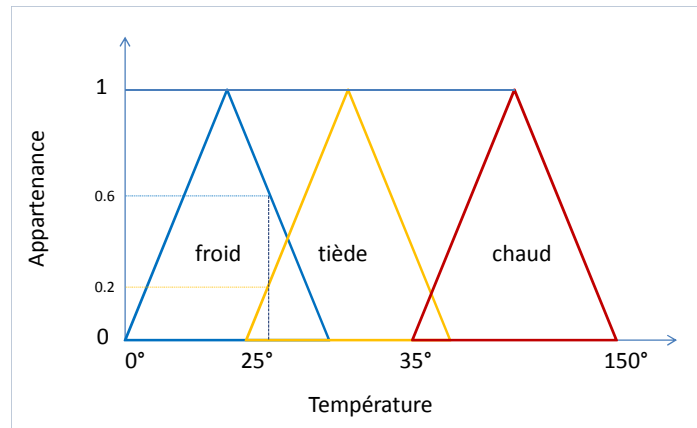


Figure 44 Ensembles flous

Un ensemble peut avoir différentes formes : triangulaire, singleton, trapézoïdal...

Théorie classique des ensembles : Si μ_A est la fonction d'appartenance de l'ensemble A

$$\forall x \in U \begin{cases} \mu_A(x) = 0 & \text{si } x \notin A \\ \mu_A(x) = 1 & \text{si } x \in A \end{cases}$$

Concept d'ensemble flou : Si μ_A est la fonction d'appartenance de l'ensemble flou A

$$\forall x \in U \mu_A(x) \in [0; 1]$$

Nous pouvons calculer le degré d'appartenance à chaque ensemble pour une température égale à 27° (trait bleu foncé en pointillé). Cette température appartient à 0.6 à l'ensemble froid, à 0.2 à l'ensemble tiède et à 0 à l'ensemble chaud.

Les opérateurs de logique floue définissent la réunion, l'intersection, le complément des ensembles flous. Pour illustrer ces opérateurs, prenons deux ensembles :

- A est l'ensemble flou de la température basse
- B est l'ensemble flou de la pression haute

OPÉRATEUR	EXEMPLE	FONCTION D'APPARTENANCE
RÉUNION	L'ensemble de la température basse OU l'ensemble de la pression haute	$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad \forall x \in U$
INTERSECTION	L'ensemble de la température basse ET l'ensemble de la pression haute	$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad \forall x \in U$
COMPLÉMENT	L'ensemble de la température NON basse	$\mu_{\bar{A}}(x) = 1 - (\mu_A(x)) \quad \forall x \in U$

Base de règles : Les systèmes à la logique floue utilisent une expertise exprimée sous forme d'une base de règles de type : **SI... ALORS...**

Exemple, si la température est basse et que la pression est haute alors ouvrir les vannes.

Inférence floue : Opération logique par laquelle on admet une proposition en vertu de sa liaison avec d'autres propositions tenues pour vraies. Si (X est A) Alors (Y est B)

- La variable floue X appartient à la classe floue A avec un degré de validité $\mu(x_0)$
- La variable floue Y appartient à la classe floue B à un degré qui dépend du degré de validité $\mu(x_0)$ de la prémisse (la partie du SI)

3.6.2.2 Etapes de construction d'un réseau de Petri flou

3.6.2.2.1 Étape 1 : Calcul des valeurs floues pour chaque variable en utilisant la fonction d'appartenance

On récolte les informations d'experts afin de construire un ensemble de règles. Tous les états et conditions du système devront être distingués. Les fonctions d'appartenance pour toutes les variables devront être définies. La figure suivante montre quelques règles, conditions, évènements et états du système.

REGLE	EVENEMENT	CONDITION	ÉTAT
R1	e1 est e11	C1	S1
R2	e1 est e12 e2 est e21	C1 et C2	S2
R3	e1 est e13	C3	S3

C1: IF (e1 est e11) THEN...

C2: IF (e2 est e21) THEN...

C3: IF (e1 est e13) THEN...

On définit un échantillon de fonction d'appartenance présenté à la figure suivante. De toute évidence, la fonction d'appartenance pour chaque variable linguistique devra être désignée par des experts.

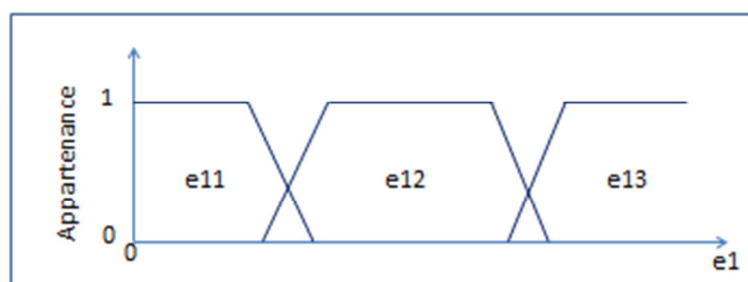


Figure 45 Définition des fonctions d'appartenances

Pour compléter cette étape, pour chaque évènement nous devrons obtenir tous les états possibles des variables linguistiques des fonctions d'appartenances et pour chaque état nous devrons avoir mis une place dans le réseau. La figure suivante montre le réseau de Petri obtenu à partir de la fonction d'appartenance $e1$ et $e2$. Il est possible de stocker la valeur de la variable

floue dans chaque place et d'ensuite connecter chaque variable linguistique d'entrée à chaque état. Tous les évènements du système doivent être présents.

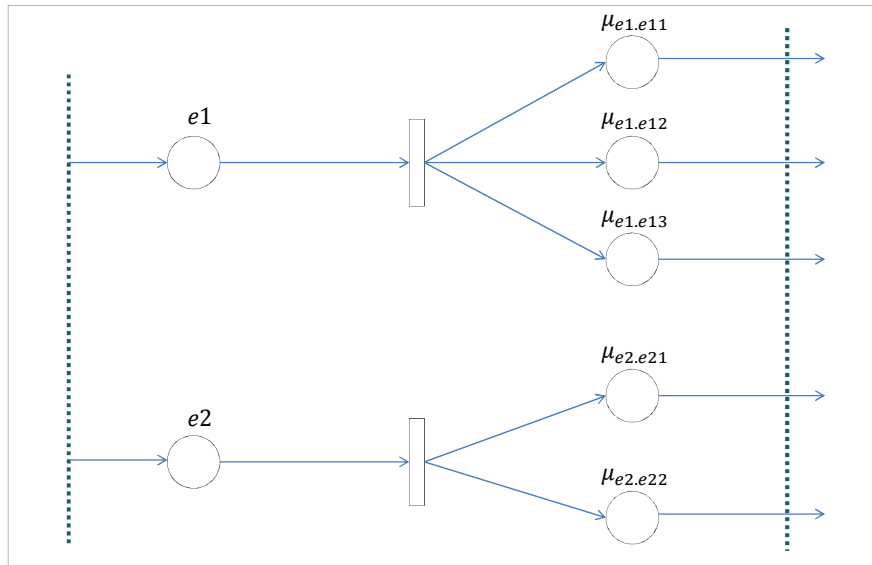


Figure 46 Calcul des valeurs floues pour chaque variable avec la fonction d'appartenance

3.6.2.2.2 Étape 2 : Calculer les règles floues de sortie et créer le réseau de Petri avec ces règles

Dans un premier temps, il faut vérifier l'exactitude des conditions. Ainsi, pour chaque condition, nous créons une transition qui valide les conditions et qui envoie les résultats dans la place suivante du réseau. Cela signifie qu'un jeton peut poursuivre son cycle de vie avec sa valeur floue. Le résultat sera donc lui aussi une valeur floue. La valeur initiale des conditions devra être créée comme une place, ensuite la transition équivalente de ces conditions devra être connectée à la place suivante.

En outre les transitions, qui ne sont pas utilisables, devront être connectées aux places finales. Pour terminer, pour chacune de ces règles, les places floues requises devront être connectées à une autre place afin de placer le résultat considéré. La figure suivante illustre ce principe.

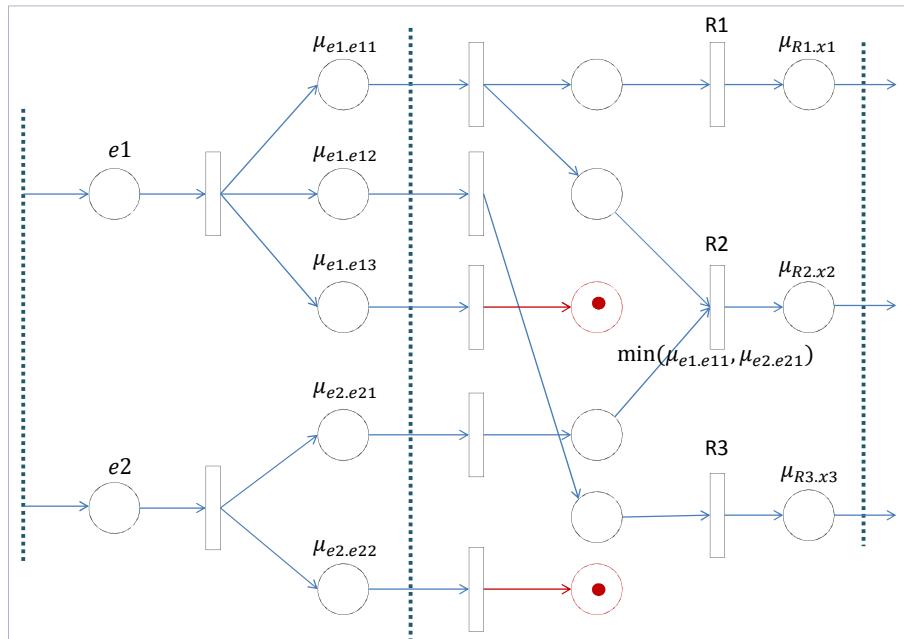


Figure 47 Calcul de la règle floue de sortie

3.6.2.2.3 Étape 3 : La combinaison des règles floues

Dans certains cas, on a plus qu'une condition. Selon la logique floue, on sélectionne la fonction appropriée. Dans cette section, on a une place pour chaque règle et chaque place a un jeton flou. Ces jetons ont deux parties : les règles floues et la valeur floue. Ici, avec l'utilisation d'une transition, le jeton qui avait une valeur floue maximum devrait être envoyé dans la place suivante. Celui-ci représente l'évènement qui avait une valeur floue.

$$OR = \mu A \cup B(x) = \max(\mu A(x), \mu B(x))$$

$$AND = \mu A \cap B(x) = \min(\mu A(x), \mu B(x))$$

La figure suivante montre les deux différentes formes. On a changé les règles et créé quelques instructions plus complexes.

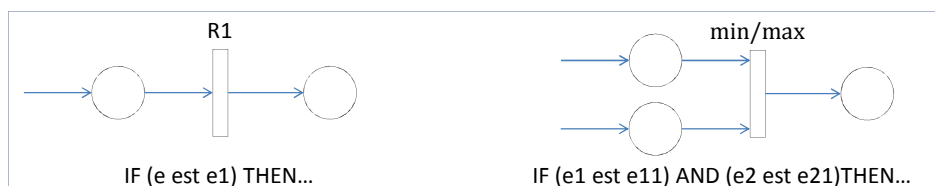


Figure 48 Illustration des deux formes

La figure suivante montre l'extrait d'une règle floue et ses valeurs avec une transition.

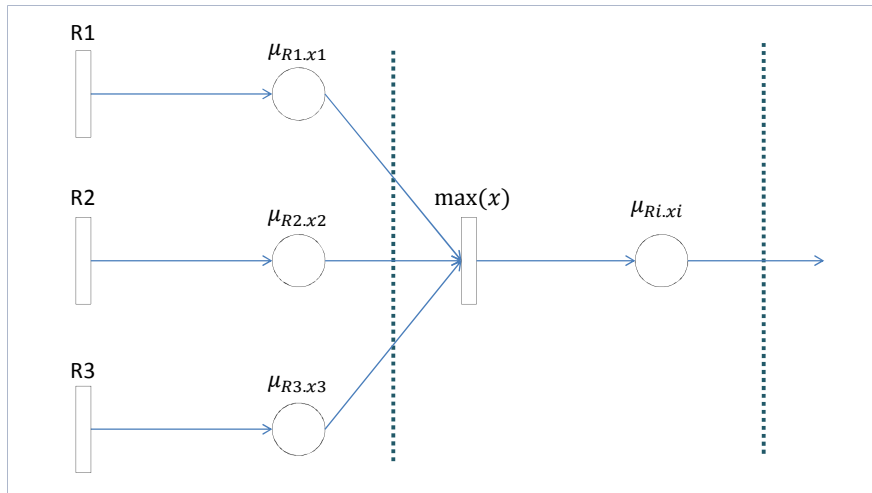


Figure 49 Illustration d'une règle floue

3.6.2.2.4 Étape 4 : Ajout de la méthode floue au réseau de Petri flou

Cette étape décrit l'état flou pour une variable d'entrée et les conséquences d'ajouts d'une méthode floue dans le réseau. Une méthode floue est ajoutée au réseau avec une transition et le résultat est aussi un état flou. Et c'est une combinaison de $\mu_{Ri.xi}$ qui signifie la règle floue et de $\mu_{methode}$, la méthode floue. Cela est calculé de la manière suivante : $\mu_{Ri.i} = (\mu_{Ri.xi} * \mu_{methode})$. Évidemment la valeur de la méthode floue peut être comprise entre 0 et 1. La figure suivante montre l'ajout de cette nouvelle étape. Aussi, si la méthode n'a pas une valeur floue, celle-ci peut être omise ou on insère la valeur 1 au lieu de la valeur floue.

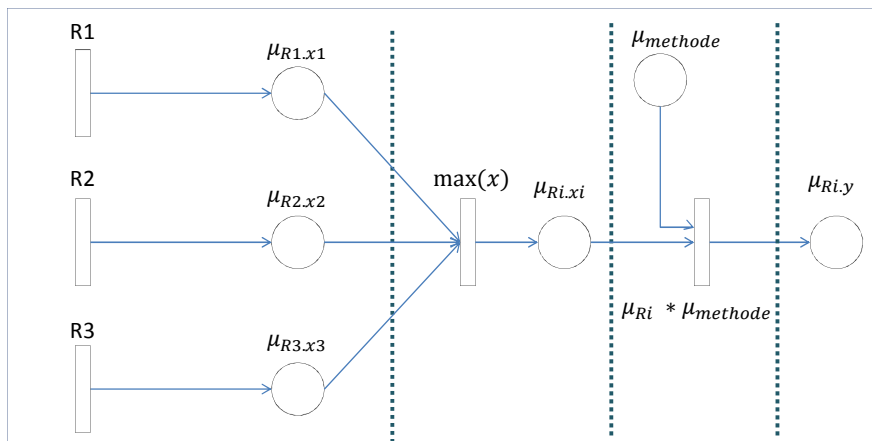


Figure 50 Réseau de Petri avec l'ajout d'une méthode floue

3.6.2.2.5 Étape 5 : Vérification de la continuité ou de la fin de la vie des jetons dans le réseau de Petri flou

Dans cette étape, on vérifie premièrement si un jeton a la capacité d'exécuter ou non. Pour cela on utilise l'opérateur OR. La figure suivante illustre cette opération.

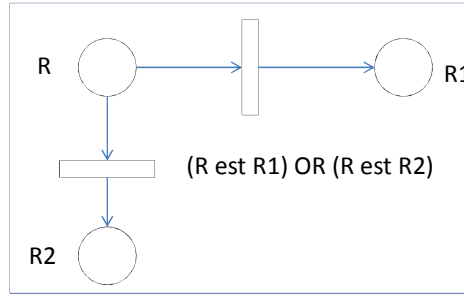


Figure 51 Illustration de l'opérateur OR

Avec l'ajout de la méthode floue dans le réseau de Petri, la variable floue peut obtenir une valeur qui ne peut pas exister dans le réseau. Donc nous devons mettre le jeton hors du réseau. Et pour montrer la fin d'un cycle de vie du jeton, on utilise la place finale comme montré à la figure suivante.

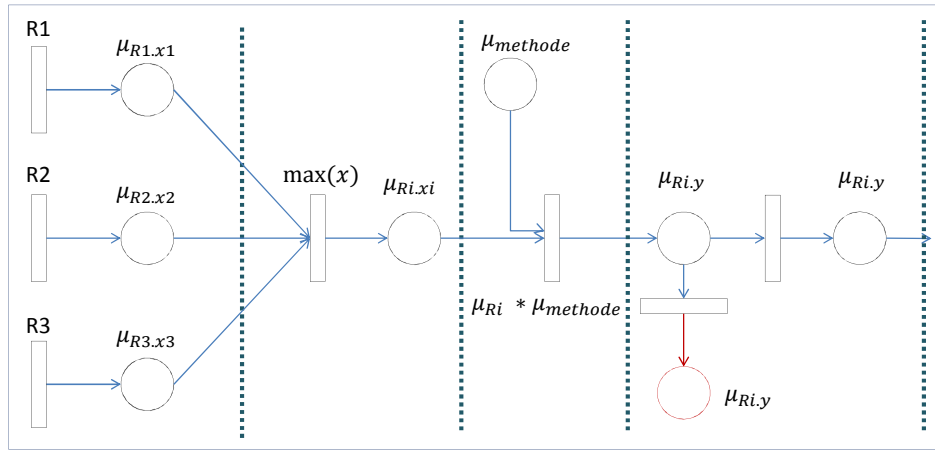


Figure 52 Vérification de la continuité et de la fin

Le résultat final du réseau devra être une valeur déterminée, donc la valeur floue devra être defuzzifiée. La défuzzification consiste à transformer une valeur floue en une valeur déterminée. On doit pour cela définir des ensembles flous de sorties dans lesquels seront insérés les différents résultats. Avec les différents points remis dans les ensembles de sorties, on applique une méthode de défuzzification, comme le centre de gravité ou le centre d'une surface.

La formule pour calculer le centre de gravité est la suivante : $COG = \frac{\int_a^b \mu_A(x) x dx}{\int_a^b \mu_A(x) dx}$

Après défuzzification, on obtient une valeur non floue qu'on situe dans l'ensemble de sortie correspondant. Cet ensemble trouvé déterminera ce que l'expert, l'opérateur devra faire.

3.6.3 Caractéristiques

Les réseaux de Petri flous ne perdent pas d'information par rapport aux réseaux de Petri ordinaires, ils augmentent le pouvoir d'expression et peuvent être analysés à l'aide de graphes d'arborescence ou d'équations.

3.7 Conclusion

Ce chapitre n'est qu'un échantillon de la série des extensions qui peuvent exister et être utilisées. Comme cité au début de ce travail, nous n'avons pu sélectionner qu'une série infime de ces extensions. Mais cette sélection nous montre une certaine diversité. Cela rend les réseaux de Petri plus adaptables à des situations. Certaines extensions présentent plus de succès, car elles sont plus faciles à utiliser. Car la difficulté d'utilisation d'une extension réside dans les connaissances supplémentaires à connaître. Des extensions peuvent, par exemple, demander des connaissances en physique, mathématique, statistique, etc. Six extensions sont présentées dans ce chapitre. Le tableau suivant résume rapidement les avantages, inconvénients et utilités de ces extensions.

EXTENSION	AVANTAGES	INCONVÉNIENTS	UTILITÉS
ARC INHIBITEUR	Facilité le test à zéro d'une place sans alourdir le réseau. Cela donne une meilleure présentation	Il n'est plus possible de vérifier les propriétés du réseau (bornée, vivant,...) ni de tester le réseau dans une matrice	Permet de tester si une place est vide. Les arcs peuvent cependant être remplacés par des places complémentaires
SYNCHRONISE	Le système a un lien avec son environnement. Approche plus réaliste des réseaux par rapport à la réalité. Conserve les propriétés (borné, vivant,...) des réseaux ordinaires	La difficulté réside dans l'identification des places/transitions et des événements. L'association des événements aux transitions peut aussi être complexe Cela peut générer un plus grand nombre d'erreurs.	Le réseau réagit par rapport aux événements de son environnement. Cette extension permet de gérer un plus grand nombre de systèmes et rajoute un pouvoir d'expression supplémentaire. Le réseau n'est plus autonome
PRÉDICAT/ TRANSITION	Ajoute la notion de variable dans le système. Conserve les propriétés (borné, vivant,...) des réseaux ordinaires	Le marquage n'est plus une distribution de constantes dans les places, mais une distribution de n-uplets. Faire attention à cette modification pour l'analyse	Permet d'introduire des variables et des opérations dans le réseau et les jetons. Approche semblable à celles des réseaux colorés.
FILES	Permet de gérer l'ordre d'arrivée et la gestion des jetons. Cela ajoute la notion de mémoire dans les réseaux.		Avec l'ajout de la gestion de l'ordre d'arrivée, cette extension est utilisée dans les systèmes pour gérer les partages des ressources

OBJET	<p>Ajout du concept « Objet ». Cela se rapproche du développement orienté objet.</p> <p>Conserve les propriétés (borné, vivant,...) des réseaux ordinaires</p>	<p>Pour l'analyse mathématique et des propriétés, faire une décomposition du réseau en classe. Cela peut devenir fastidieux.</p> <p>Demande de différencier les classes du système avant de développer le réseau de Petri</p>	<p>L'ajout du concept Objet permet de modéliser un plus grand nombre de systèmes actuels.</p>
FLOU	<p>L'ajout de la logique floue permet au réseau d'être plus proche de la réflexion humaine.</p> <p>Conserve les propriétés (borné, vivant,...) des réseaux ordinaires</p>	<p>Exige du modélisateur des connaissances supplémentaires dans la logique floue et une identification des règles d'inférences</p>	<p>Cette extension est forte utile dans la maintenance industrielle, car les réseaux contiennent des diagnostics plus humains.</p>

CHAPITRE 4 : Les réseaux colorés

4.1 Propriété manquante aux réseaux de Petri

Dans un réseau de Petri ordinaire, les jetons ont uniquement un rôle de compteur. Afin de comprendre la nouvelle extension qui va être introduite, regardons la figure suivante.

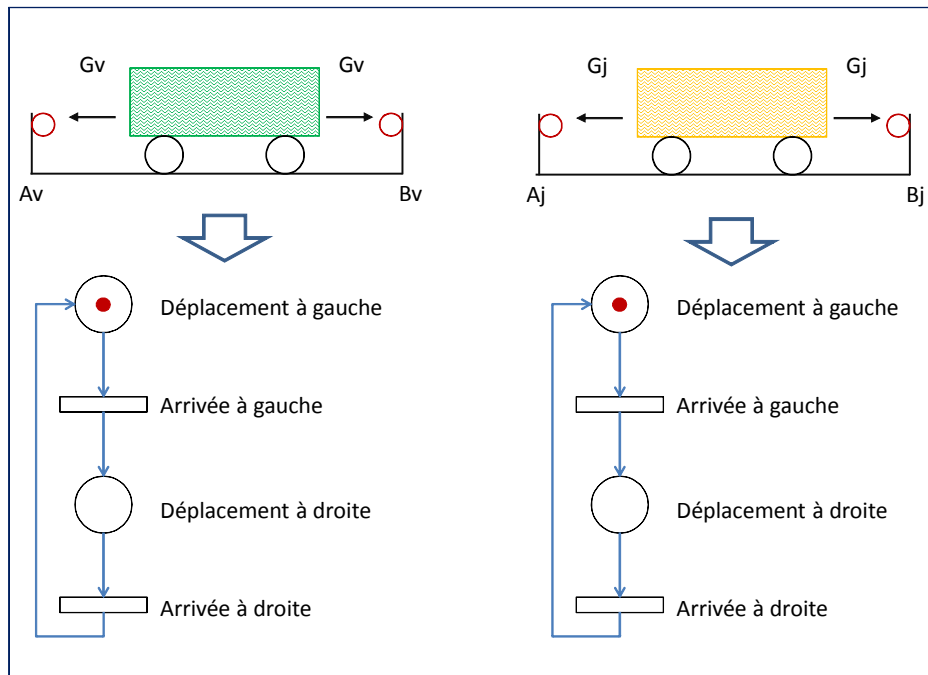


Figure 53 Modélisation d'un système de chariot

Ce système est composé de deux chariots qui exécutent la même série de mouvements. Cependant, les chariots sont différents. Il y a un chariot vert et un chariot jaune. Un réseau de Petri a été réalisé pour chaque chariot. Mais ceux-ci sont similaires. Il y a une redondance d'information. La modélisation de ce type système peut rapidement devenir laborieuse. Car si les réseaux grandissent, ils peuvent devenir illisibles.

Modéliser le système par un seul réseau reste une alternative. Cependant, les chariots ne sont plus identifiables : il y a une perte d'information et d'identité. En effet, les jetons n'ayant qu'un rôle de compteur, il n'est plus possible de connaître le jeton qui identifie le chariot vert ou jaune.

La figure suivante reprend les deux réseaux de l'illustration précédente, en rassemblant les différents jetons dans un même réseau. En effet, il est impossible, dans cette situation, de différencier les deux chariots.

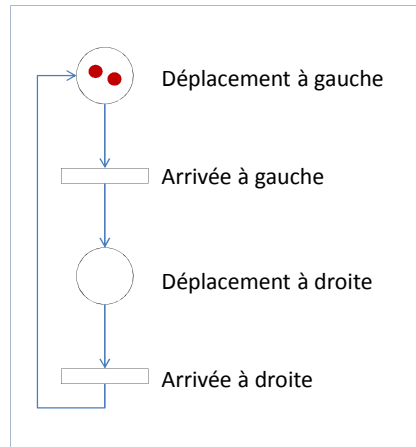


Figure 54 Rassemblement des deux réseaux de Petri ordinaires

Pour remédier à ce problème, une solution est d'attribuer une partie de l'information aux jetons. Ce qui permet d'identifier les jetons. Les valeurs des jetons s'appellent la **couleur** de jeton et les types de données s'appellent **l'ensemble des couleurs**. On parle maintenant de jetons colorés. Dans la figure suivante, il y a un jeton vert pour le chariot vert et un jeton jaune pour le chariot jaune.

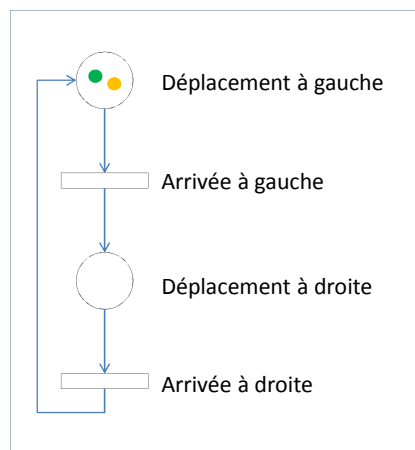


Figure 55 Identification de jeton par une couleur

L'avantage de cette extension est d'alléger la représentation graphique des réseaux de Petri ordinaires. Elles sont surtout utilisées pour modéliser des protocoles de communications, des systèmes distribués ou des systèmes de productions.

4.2 Définition

4.2.1 Les réseaux colorés simples

Avant de donner la définition formelle des réseaux colorés, il est nécessaire d'expliquer quelques notations. On définit, pour un réseau, un ensemble des couleurs qui pourra être attribué aux jetons. Dans l'exemple précédent, l'ensemble des couleurs est le jaune et le vert.

Définition : Soit $C = \{c_1, c_2, \dots\}$ un ensemble fini de couleurs. Un marquage M des places d'un réseau coloré se définit comme une application de l'ensemble des places P dans l'ensemble des fonctions de C dans les entiers : $M: P \rightarrow \mathbb{N}^C$. Ainsi $M(p)(c_i)$ désigne le nombre d'éléments de couleur c_i au sein du marquage $M(p)$ de la place p .

Les places d'un réseau coloré peuvent contenir des marques colorées. Plusieurs marques de la même couleur peuvent se trouver dans la même place. Par exemple, le marquage de la place « déplacement gauche » de la figure précédente est : $1\langle\text{jaune}\rangle + 1\langle\text{vert}\rangle$. Si un jeton $\langle\text{jaune}\rangle$ s'ajoute à la place, le marquage devient $2\langle\text{jaune}\rangle + 1\langle\text{vert}\rangle$.

Pour faire circuler les jetons dans le réseau, on associe un ensemble de couleurs à chaque transition. Chaque couleur représente une possibilité distincte de franchissement. Dans la figure suivante, la transition « Arrivée à gauche » accepte de laisser passer des jetons verts et jaunes. La transition « Arrivée à droite » accepte, quant à elle, les jetons vert, jaune et rouges. Les jetons actuels du réseau vont pouvoir circuler dans le réseau en toute liberté. Cependant, si un jeton rouge est inséré dans la place « Déplacement à gauche », il ne pourra pas franchir la transition « Arrivée à gauche »

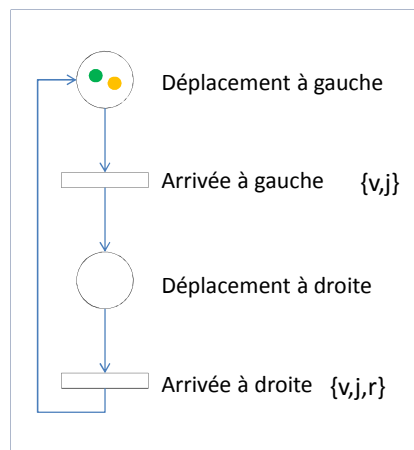


Figure 56 Association des couleurs aux transitions

Le poids des arcs est remplacé par des expressions spécifiant l'utilisation de marques colorées. Les expressions sont de deux types :

- $\langle K \rangle$ où K désigne un **sous-ensemble quelconque de couleurs**. Celui-ci permet de lier le franchissement d'une transition à la présence d'un sous-ensemble K de couleurs au sein du marquage d'une place d'entrée de cette transition. En d'autre mot, K désigne toutes les couleurs qui doivent être présentes dans une place d'entrée.
- $\langle x \rangle$ où x est une **variable libre prenant ses valeurs dans un sous-ensemble de couleurs** spécifiées par l'énoncé « *domaine* (x)... » ; un tel énoncé faisant partie de la description d'un réseau coloré. Celui-ci permet d'induire une condition de franchissement soumise à la présence d'une même couleur au sein du marquage de toutes les places d'entrée d'une transition, marquée à cette dernière par des arcs étiquetés par la même variable.

La figure suivante illustre $\langle K \rangle$ dans le réseau (a) et $\langle x \rangle$ dans le réseau (b). Nous changeons les couleurs des transitions. Elles permettent de laisser passer les couleurs vert, jaune, rouge et bleu.

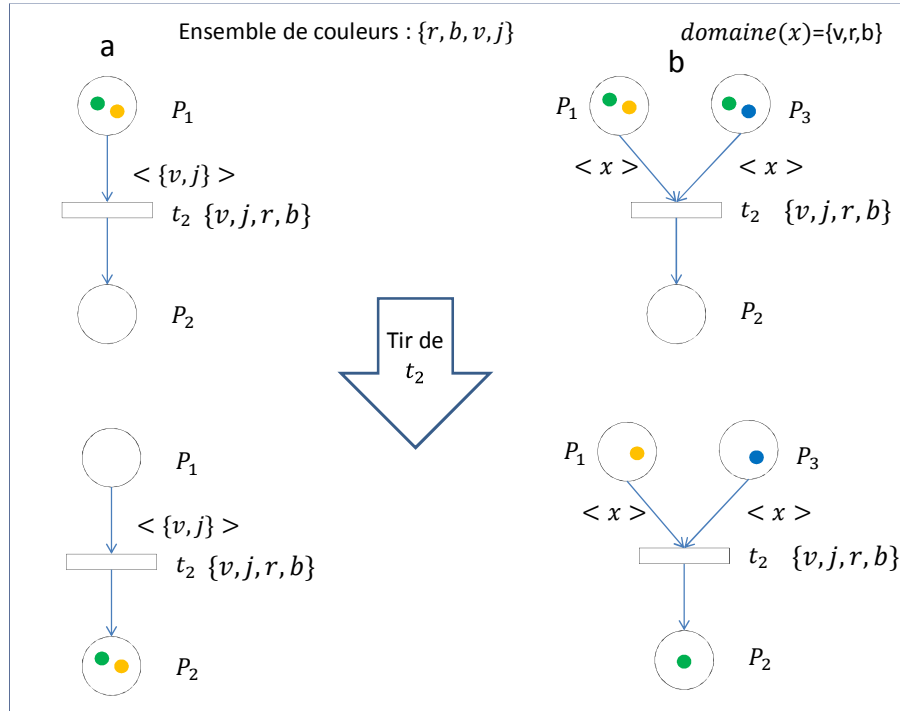


Figure 57 Réseau avec expressions des arcs

Avant de donner la règle de franchissement, on peut constater que dans le réseau de gauche, la transition t_2 est tirable, car il y a un jeton vert et un jeton jaune. Les deux jetons seront retirés de la place P_1 pour être mis dans la place P_2 .

Par contre dans le réseau de droite, il n'y a que le jeton vert qui est commun aux deux places d'entrée. Ces jetons verts seront les seuls prélevés des places d'entrée. Il ne restera dans P_1 qu'un jeton jaune et dans P_3 un jeton bleu.

Le franchissement d'une transition t depuis un marquage M est régi par les règles suivantes :

Définition : t est franchissable depuis M ($M(t >)$) si et seulement si

- Pour tout sous-ensemble de couleurs K , pour toute place p liée à t par un arc (p, t) portant l'étiquette $\langle K \rangle$, pour toute couleur $s \in K$ $M(p)(s) \geq 1$
- Pour toute variable x , il existe $s \in \text{domaine}(x)$, tel que pour toute place p' liée à t par un arc (p', t') d'étiquette $\langle x \rangle$, $M(p')(s) \geq 1$.

Illustrons cette nouvelle règle par un petit exemple en sachant que $C = \{r, v, b, j\}$, $\text{domaine}(x) = \{r, v\}$ et $\text{domaine}(y) = \{v, b, j\}$.

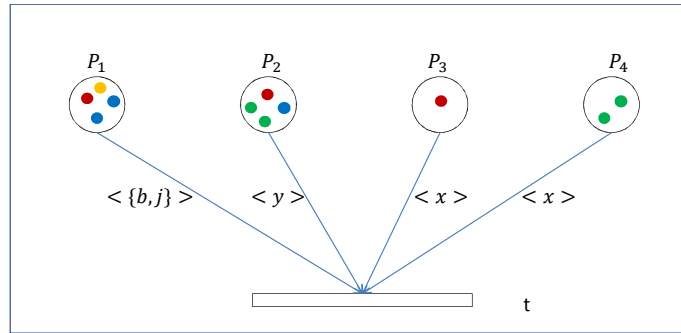


Figure 58 Transition non valide d'un réseau coloré

Regardons la figure ci-dessus, la transition t n'est pas franchissable depuis ce marquage initial, car P_3 et P_4 ne contiennent pas une même couleur. En effet, P_3 ne contient qu'un jeton rouge et P_4 ne contient que des jetons vers. Or, la transition exige un jeton rouge ou un jeton vert. Il n'y a aucune couleur en commun entre ces deux places. La règle de franchissement n'est pas respectée.

Par contre, à la figure suivante, la transition est franchissable. Car P_3 et P_4 contiennent chacun des jetons verts. P_1 a un jeton jaune et un jeton bleu et P_2 a au moins un jeton vert et un jeton bleu.

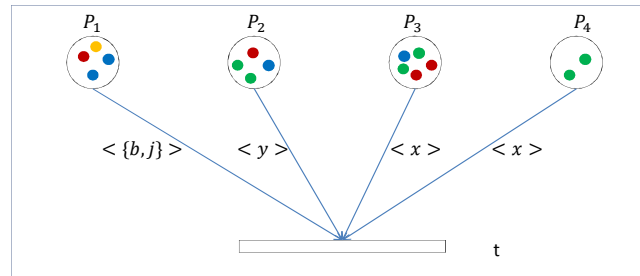


Figure 59 Transition valide d'un réseau coloré

Si le franchissement de la transition est valide pour un sous-ensemble, il suffit ensuite d'ajouter ou retrancher ce sous-ensemble au marquage des places d'entrée et de sortie. Pour définir le marquage résultant d'un franchissement, nous pouvons étendre la notion de couleur effective pour des variables qui ne sont pas étiquettes des arcs entrants de t :

$EF(x, M, t)$ désigne l'ensemble des couleurs effectives de x pour t au marquage M si il existe une place p et un arc (p, t) d'étiquette $\langle x \rangle$; sinon $EF(x, M, t) = \text{domaine}(x)$.

Pour illustrer la notion de couleur effective, reprenons les exemples précédents : le vert et le bleu sont des couleurs effectives de y pour les deux exemples. Le vert est une couleur effective de x pour tout t au marquage du deuxième exemple. Le marquage M' se définit comme suit :

Définition : Nous avons $M(t > M')$ si et seulement si M' satisfait les trois conditions suivantes :

1. Pour toute place p qui n'est pas connectée à t , $M'(p) = M(p)$
2. Pour tout sous-ensemble $K \subseteq C$, pour toute place p liée à t par un arc v d'étiquette $\langle K \rangle$
 - Si $v = (p, t)$ alors pour toute couleur $s \in C$

$$M'(p)(s) = \begin{cases} M(p)(s) - 1 & \text{si } s \in K \\ M(p)(s) & \text{sinon} \end{cases}$$

- Si $v = (t, p)$ alors pour toute couleur $s \in C$

$$M'(p)(s) = \begin{cases} M(p)(s) + 1 & \text{si } s \in K \\ M(p)(s) & \text{sinon} \end{cases}$$

3. Pour toute variable x , il existe $s \in EF(x, M, t)$ tel que pour toute place p liée à t par un arc v d'étiquette $< x >$:

- Si $v = (p, t)$ alors pour toute couleur $s' \in C$

$$M'(p)(s') = \begin{cases} M(p)(s') - 1 & \text{si } s' = s \\ M(p)(s') & \text{sinon} \end{cases}$$

- Si $v = (t, p)$ alors pour toute couleur $s' \in C$

$$M'(p)(s') = \begin{cases} M(p)(s') + 1 & \text{si } s' = s \\ M(p)(s') & \text{sinon} \end{cases}$$

Il faut signaler que le marquage obtenu par le franchissement d'une transition n'est pas toujours unique. Car cet ensemble des valeurs effectives d'une variable libre peut être de cardinalité supérieure à 1. Illustrons cela par un exemple. Nous reprendrons donc la figure précédente en y ajoutant trois places de sorties à la transition t avec :

- $\text{domaine}(x) = \{r, v\}, \text{domaine}(y) = \{v, b, j\}, \text{domaine}(z) = \{r, j\}$

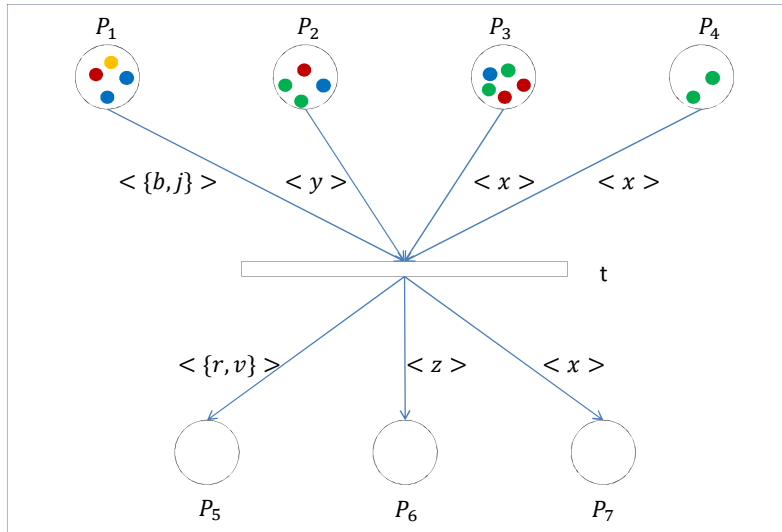


Figure 60 Ajout de places de sortie au réseau coloré précédent

Comme dans l'exemple suivant, la transition est franchissable pour ce marquage. Les ensembles de valeurs effectives des variables x, y, z sont :

$$\begin{aligned} EF(x, M, t) &= \{v\} \\ EF(y, M, t) &= \{v, b\} \\ EF(z, M, t) &= \{r, j\} \end{aligned}$$

Il y aura quatre marquages accessibles par le franchissement de t : M'_1, M'_2, M'_3, M'_4 avec

	M'_1	M'_2	M'_3	M'_4
p_1	$\langle 1,0,1,0 \rangle$	$\langle 1,0,1,0 \rangle$	$\langle 1,0,1,0 \rangle$	$\langle 1,0,1,0 \rangle$
p_2	$\langle 1,1,1,0 \rangle$	$\langle 1,1,1,0 \rangle$	$\langle 1,2,0,0 \rangle$	$\langle 1,2,0,0 \rangle$
p_3	$\langle 2,1,1,0 \rangle$	$\langle 2,1,1,0 \rangle$	$\langle 2,1,1,0 \rangle$	$\langle 2,1,1,0 \rangle$
p_4	$\langle 0,1,0,0 \rangle$	$\langle 0,1,0,0 \rangle$	$\langle 0,1,0,0 \rangle$	$\langle 0,1,0,0 \rangle$
p_5	$\langle 1,1,0,0 \rangle$	$\langle 1,1,0,0 \rangle$	$\langle 1,1,0,0 \rangle$	$\langle 1,1,0,0 \rangle$
p_6	$\langle 1,0,0,0 \rangle$	$\langle 0,0,0,1 \rangle$	$\langle 1,0,0,0 \rangle$	$\langle 0,0,0,1 \rangle$
p_7	$\langle 0,1,0,0 \rangle$	$\langle 0,1,0,0 \rangle$	$\langle 0,1,0,0 \rangle$	$\langle 0,1,0,0 \rangle$

Pour conclure cette section, nous pouvons constater, avec le tableau précédent, qu'il est possible de construire un réseau ordinaire. Cette manipulation est expliquée dans la section 4.4 « Dépliage ». Nous pouvons par contre remarquer, dans l'exemple précédent, qu'il existe $\text{domaine}(x) \times \text{domaine}(y) \times \text{domaine}(z) = 12$ possibilités de franchissements. La puissance d'abréviation des réseaux colorés est telle qu'elle permet d'aborder la modélisation de systèmes en vraie grandeur.

4.2.2 Les réseaux colorés complexes

Les réseaux colorés complexes sont les plus utilisés, car ils permettent de rajouter un plus grand nombre de conditions sur les arcs. Dans les réseaux colorés complexes, les poids des arcs sont remplacés par des fonctions (dans la figure suivante, elles sont définies par f). Elles établissent une correspondance entre chaque couleur de la transition et les couleurs de la place. Par exemple, pour un arc v liant une place p à une transition t , il peut y avoir quatre fonctions définies (une par couleur : noir, jaune, rouge, blanc). Il peut avoir aussi des fonctions sur les arcs de sorties. Celles-ci peuvent éventuellement modifier les couleurs des jetons.

Dans l'exemple suivant, si nous avons la fonction $f(j) = r + n$, cela indique qu'il faut, pour appliquer la fonction $f(j)$, un jeton rouge et un jeton noir. La transition est franchie. Nous reprenons la fonction de la même couleur $g(j)$. Elle indique qu'il faut mettre, dans la place de sortie, un jeton bleu.

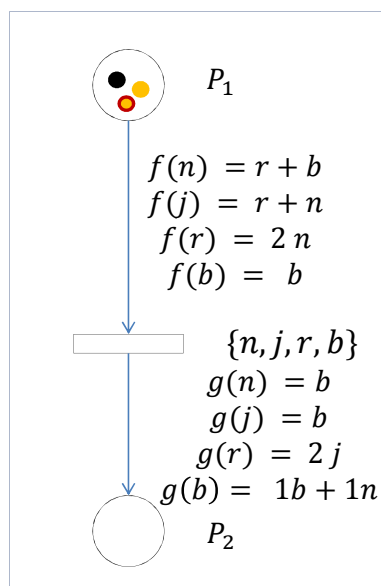


Figure 61 Attribution des fonctions à l'arc

Définition : On appelle **multi-ensemble** d'éléments de A toute application de A dans \mathbb{N} . On note $\mathcal{M}(A)$ l'ensemble des multi-ensembles de A .

Soit $A = \{a, b, c, d\}$. $E = \{a, a, b, c, c, c\}$ est un multi-ensemble de A .

On le note aussi $E = 2a + b + 3c$

On note $\mathcal{P}(A)$ l'ensemble des parties de A .

Donnons la définition formelle d'un réseau coloré complexe.

Définition : Un réseau coloré est un septuplé $(P, T, Pré, Post, C, C_T, C_P)$ où :

- P est un ensemble fini de places
- T est un ensemble fini de transitions disjoint de P
- C est un ensemble fini de couleurs
- $C_T: T \rightarrow \mathcal{P}(C)$
- $C_P: P \rightarrow \mathcal{P}(C)$
- $Pré$ et $Post$ sont respectivement les fonctions de précondition et postcondition et vérifient :
 - $Pré(p, t): C_T(t) \rightarrow \mathcal{M}(C_P(p))$
 - $Post(t, p): C_T(t) \rightarrow \mathcal{M}(C_P(p))$

Un marquage est une application $P \rightarrow \mathcal{M}(C)$ tel que $\forall p \in P, M(p) \in \mathcal{M}(C_P(p))$.

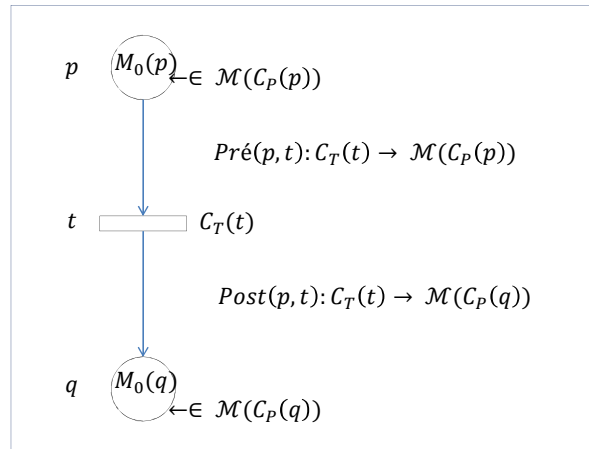


Figure 62 Réseau coloré complexe

Pour franchir une transition t par rapport à une couleur, on sélectionne les jetons spécifiés dans la fonction de précondition. Si tous les jetons demandés sont présents, on soustrait ces jetons à la place d'entrée. On ajoute ensuite dans la place de sortie, les jetons définis dans la fonction de sortie, par rapport à la couleur de franchissement.

La figure suivante illustre ce franchissement de t par rapport à la couleur jaune. Celle-ci demande un jeton rouge et un jeton noir. On ajoute ensuite, dans la place de sortie, trois jetons noirs.

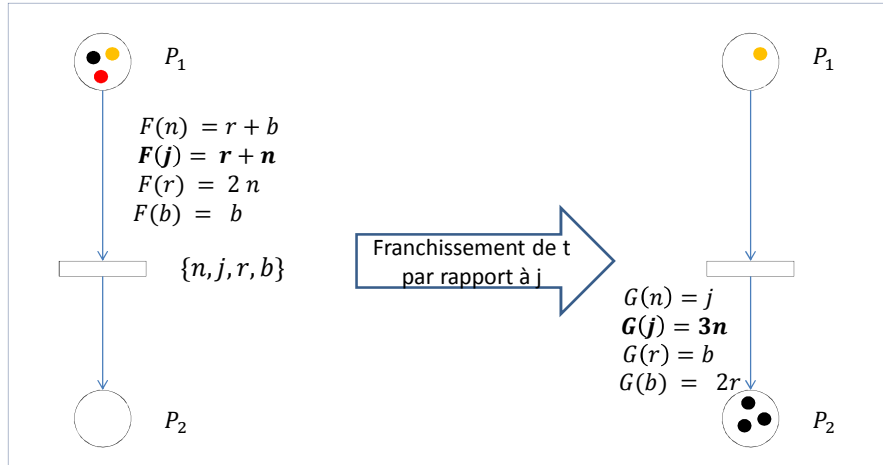


Figure 63 Franchissement d'une transition par rapport à une couleur

Définition :

- Une transition t est **valide** par rapport à la couleur c ($c \in C_T(t)$) à partir d'un marquage M si et seulement si pour toute place p de P , $\text{Pré}(t, p)(c) \leq M(p)(c)$.
- Le **tir** de t par rapport à la couleur c depuis M conduit au marquage M' défini par : $\forall p \in P, \forall c \in C_p, M'(p)(c) = M(p)(c) - \text{Pré}(t, p)(c) + \text{Post}(p, t)(c)$.
- Une **séquence franchissable** est une suite de couples $(u_1, c_1)(u_2, c_2) \dots (u_k, c_k)$ où $u_i \in T$ et $c_i \in C_T(u_i)$

En d'autres mots, si une transition T est validée pour une couleur c_k , alors la transition peut être franchie. On le note T/c_k .

Pour bien comprendre le concept de franchissement ; voici un deuxième exemple. À la figure suivante, lorsqu'on franchit la transition t par rapport à la couleur $vert(v)$, on retire un jeton *bleu*(b), un jeton *vert*(v) et un jeton *jaune*(j) de p . On ajoute un jeton *rouge*(r), deux jetons *bleu*(b) et un jeton *jaune*(j) à q .

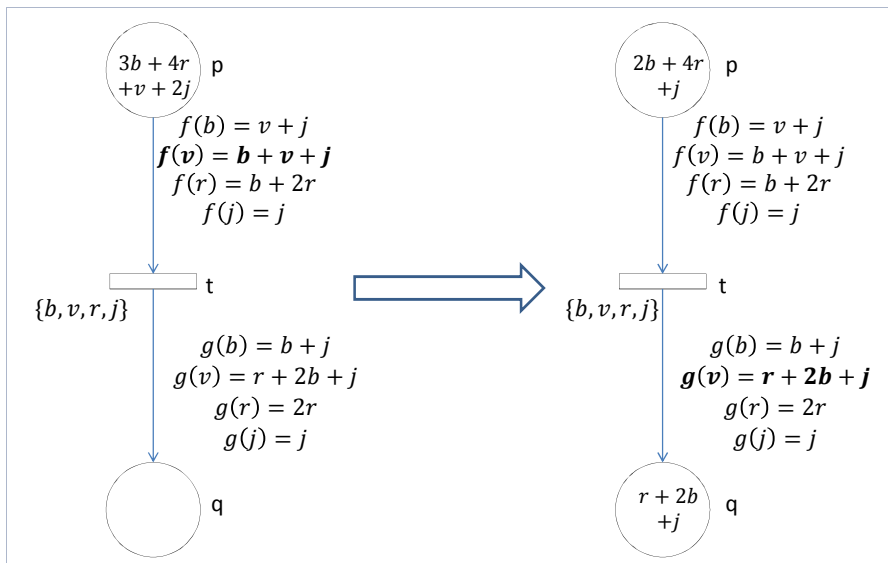


Figure 64 Comportement d'un réseau coloré

Une dernière particularité. Si nous souhaitons retenir dans un jeton plusieurs types d'informations. Par exemple : une machine produit deux types de pièces a et b . Ces pièces peuvent être soit dans la machine(m), soit dans le stock(s).

Nous voudrions connaître le type et la situation de la pièce. La couleur du jeton est définie par un doublet $\langle ., . \rangle$. Par exemple, la pièce a , qui est dans la machine, s'écrit $\langle a, m \rangle$. On parle de **couleur complexe**.

4.3 Caractéristiques

Les réseaux de Petri colorés conservent les mêmes propriétés que les réseaux de Petri ordinaires. Mais ceux-ci doivent tenir compte des couleurs, car ils ne sont qu'une représentation des réseaux ordinaires. Les propriétés peuvent toutefois se présenter de façons différentes.

4.3.1 Matrice d'incidence

L'élément (i, j) de la matrice d'incidence est égal à la différence des deux fonctions $Post(P_i, T_j)$ et $Pré(P_i, T_j)$

4.3.2 Réseau borné

Une place P_i est bornée pour C_x si quel que soit le marquage de P_i , atteignable à partir de M_0 , il existe B_{ix} tel que le nombre de jetons de couleur C_x soit inférieur ou égal à B_{ix}

Un réseau coloré est borné si chaque place est bornée pour toutes les couleurs C_x

4.3.3 Vivacité

Une transition T_i est vivante pour M_0 et pour C_x si quelque soit le marquage M atteignable à partir de M_0 , il existe une séquence S qui part de M et qui franchit T_i pour C_x

Un réseau coloré est vivant si toutes ses transitions sont vivantes pour toutes les couleurs de franchissement.

On peut aussi dire qu'un réseau coloré est vivant, si son réseau de Petri ordinaire déplié est vivant (voir la section sur le dépliage).

4.3.4 Blocage

Un réseau coloré est sans blocage si pour tout marquage accessible, il y a au moins une transition franchissable pour au moins une de ses couleurs.

4.4 Dépliage

Les réseaux de Petri colorés ont la même puissance de description qu'un réseau de Petri ordinaire tout en étant plus compacts. Les réseaux colorés n'offrent pas de puissance d'expression supplémentaire, ils ne sont qu'une abréviation des réseaux ordinaires.

Ceci implique que l'on peut modéliser un système par un réseau coloré et par un réseau ordinaire. Il est possible de passer de l'un à l'autre. Cette possibilité de conversion s'appelle le **dépliage**. Les caractéristiques des réseaux colorés sont donc les mêmes que celles des réseaux

ordinaires, car il n'y a pas de modification d'expression dans les réseaux colorés. En effet, un réseau ordinaire peut être vu comme un réseau coloré, mais à une seule couleur.

Pour déplier un réseau, celui-ci doit passer par une série d'étapes :

- Dupliquer les places : on duplique chaque place autant de fois qu'elle peut contenir de couleur.

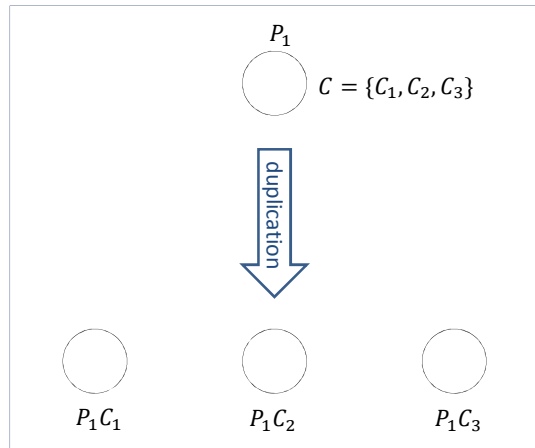


Figure 65 Duplication des places

- Marquage initial : on reporte dans chaque place, les jetons de la couleur qui lui correspondent, s'ils existent dans la place initiale pour M_0 .

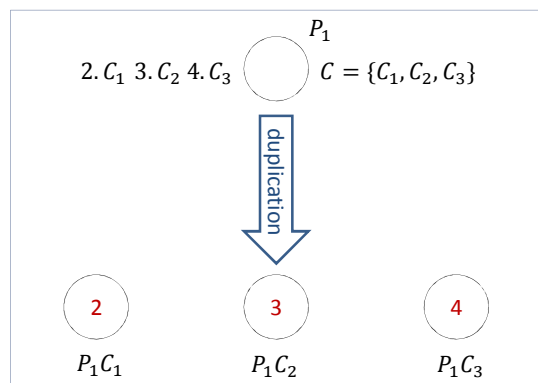


Figure 66 Marquage initial

- Dupliquer les transitions : on duplique chaque transition autant de fois qu'elle a de couleur de franchissement.

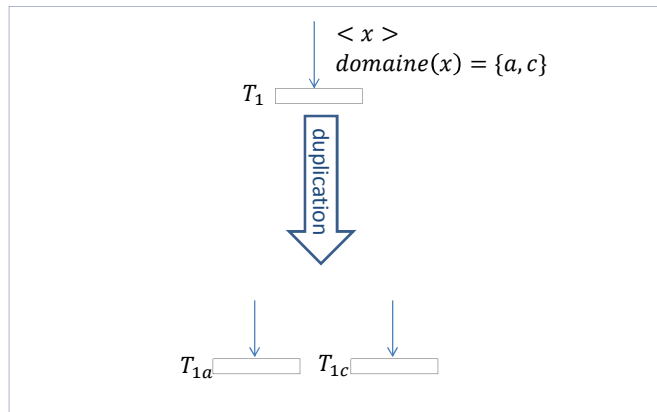


Figure 67 Duplication des transitions

- Lier les places et les transitions : On relie les places aux transitions.

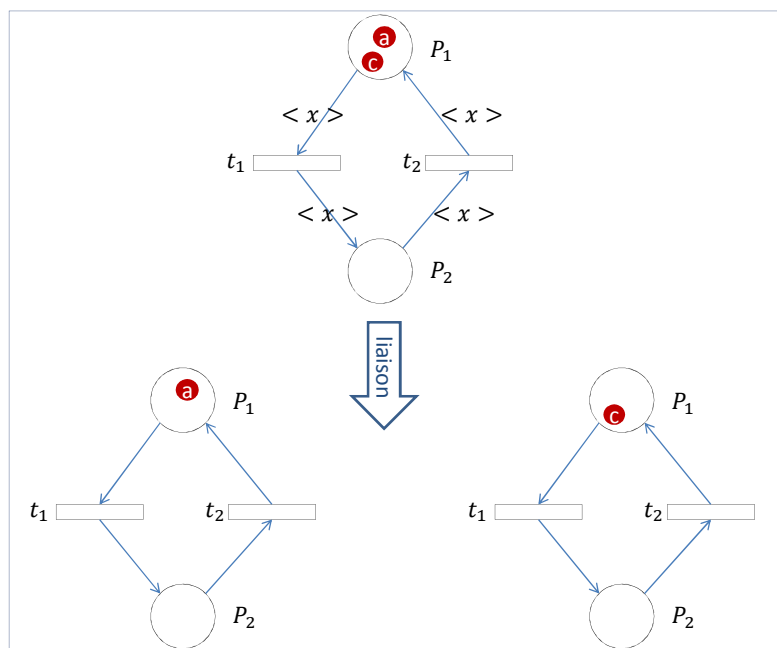


Figure 68 jonctions entre les places et les transitions

Prenons un petit exemple pour rassembler toutes ces indications. Nous avons le réseau coloré suivant :

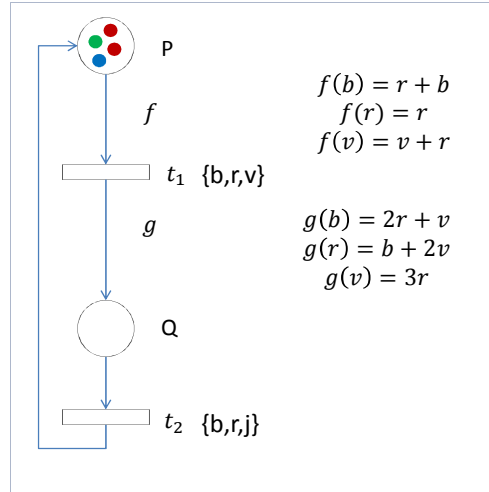


Figure 69 Réseau coloré avant dépliage

Nous avons $C_P = C_Q = C_{t_1} = C_{t_2} = C = \{b, r, v\}$. On remplace P et Q , chacune par trois places (une place par couleur). Les transitions t_1 et t_2 sont aussi remplacées chacune par trois transitions (à nouveau une par couleur). Les valuations des arcs sont obtenues par projection des valuations des arcs du réseau coloré sur la bonne couleur. La figure suivante est le réseau ordinaire obtenu par dépliage.

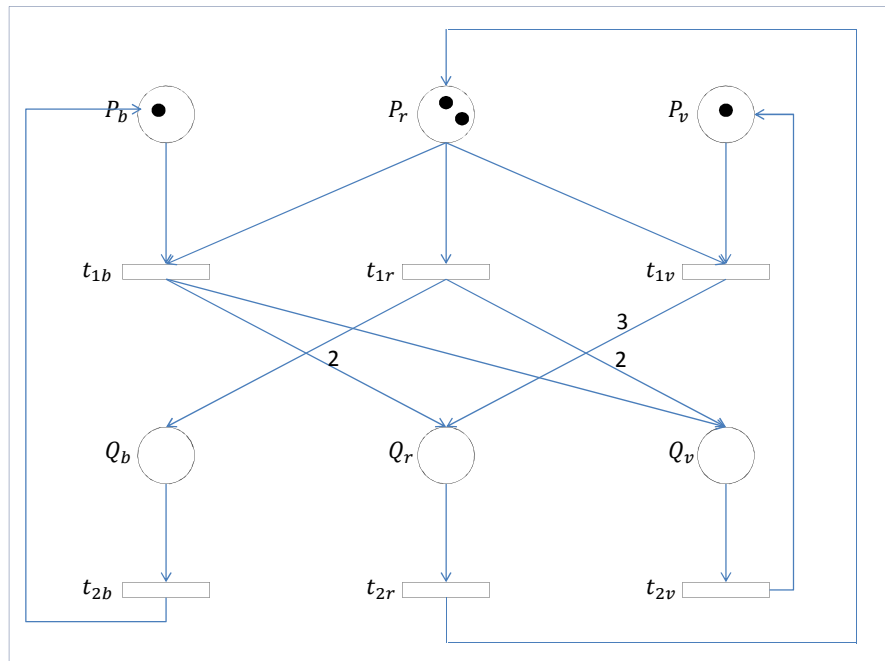


Figure 70 Dépliage du réseau coloré

L'opération peut être faite dans l'autre sens, c'est le **pliage**. Il suffit d'inverser les opérations.

Concernant les propriétés des réseaux, nous pouvons constater que le réseau ordinaire est vivant. Cela implique que le réseau coloré est lui aussi vivant. Les propriétés sont gardées.

CHAPITRE 5 : Les réseaux temporisés

5.1 Propriété manquante aux réseaux de Petri

Dans les réseaux de Petri ordinaires, le temps n'est pas pris en compte. Il n'apparaît pas dans le modèle. Or dans certains systèmes, il est important d'évaluer le temps d'exécution d'une série d'opérations.

Prenons comme exemple la cuisson d'un œuf à la coque. Les étapes de cette recette et le réseau de Petri ordinaire sont donnés ci-dessous.

1. Porter de l'eau à ébullition
2. Lorsque l'eau bout, plonger les œufs dans l'eau bouillante
3. Laisser cuire les œufs 3 minutes
4. Retirer les œufs de l'eau avec une grande cuillère.



À travers cette description, nous pouvons identifier les places et les transitions

Places	N° des places	Transitions	N° des transitions
Eau froide	P_1	Porter à ébullition	T_1
Eau bouillante	P_2	Ajouter œuf dans l'eau	T_2
Œuf dans l'eau	P_3	Retirer œuf de l'eau	T_3
Œuf prêt	P_4		

Avec les

nt.

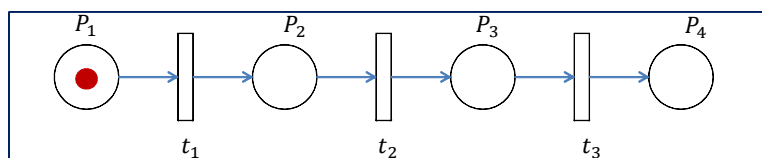


Figure 71 Réseau ordinaire d'une recette d'œuf à la coque

Toutefois, le réseau de la figure précédente ne respecte pas la totalité de la recette. Il ne modélise pas le temps de cuisson de l'œuf. On insère dans le réseau une durée de 3 minutes sur la place P_3 . Cette durée bloque le jeton dans la place le temps de la cuisson. La figure suivante illustre ce

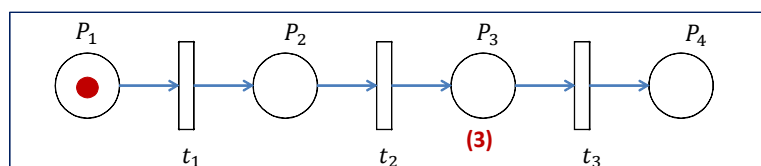


Figure 72 Insertion de la durée sur une place

Cette nouvelle extension permet de bloquer, pendant un certain laps de temps, un jeton dans une place. On intègre explicitement la durée des activités dans le modèle, pour évaluer le temps d'exécution d'une série d'opérations. Ce type d'extension est par exemple utilisé dans les chaînes de fabrications industrielles.

5.2 Définition

Les réseaux temporisés permettent simplement de bloquer un jeton dans une place ou une transition pendant une durée définie. Cela modélise le temps d'attente ou le temps d'exécution d'une action dans la vie réelle.

Les jetons peuvent être bloqués dans les transitions (**réseaux T-temporisés**) ou dans les places (**réseaux P-temporisés**). Cela veut dire simplement que la durée est soit fixée sur les transitions, soit sur les places.

Si la temporisation est portée sur les places, ces dernières modélisent les activités. Les transitions modélisent le passage d'une activité à une autre. Ce passage est **instantané**.

Et inversement, si la temporisation est portée sur les transitions. La figure suivante illustre les deux modes de temporisations.

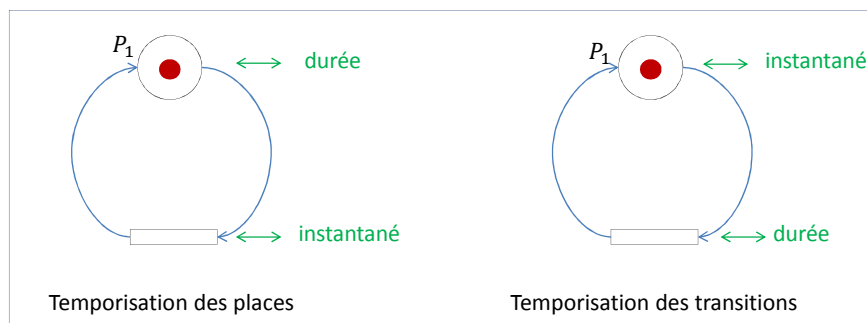


Figure 73 Les deux modes de temporisations

La figure suivante nous présente un exemple de réseau temporisé (il est même T-temporisé) pour comprendre la notion de blocage durant un temps $\delta(U)$.

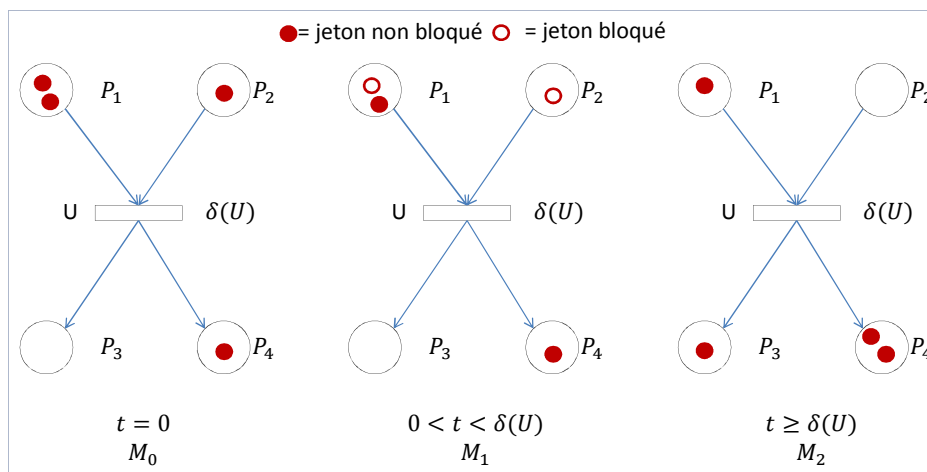


Figure 74 Franchissement d'une transition temporisée

5.2.1 Les réseaux P-temporisés

Les réseaux P-temporisés sont les réseaux dans lesquels on associe une durée à chaque place. La marque arrivant dans la place doit y rester au moins le temps spécifié par la fonction de temporisation. Dans l'exemple précédent, le jeton doit rester 3 minutes au moins. Après ce temps d'attente, la transition en sortie de la place pourra être tirée.

Définition : Un réseau de Petri P-temporisé est défini par le doublet $\langle N, d \rangle$ où :

- N est un réseau places/transitions
- $d: P \rightarrow \mathbb{N}$ est la fonction de temporisation

La règle de tir d'une transition doit tenir compte du temps. Soit t_0 l'instant où un jeton est introduit dans une place p . Le jeton est indisponible (il est **gelé**) pendant une durée égale à $d(p)$ c'est-à-dire dans l'intervalle $[t_0, t_0 + d(p)]$. Le jeton pourra être utilisé par le tir d'une transition de sortie après la durée définie. Nous notons les temporisations à côté des places (parenthèses).

La figure suivante illustre les réseaux P-temporisés : Dans le premier réseau, la marque est déposée dans P_1 , mais elle est gelée durant un temps d .

t_2 n'est pas franchissable. Au bout de d unités de temps, la marque est dégelée. La transition t_2 devient tirable après un temps non déterminé (représenté par le signe « ? »).

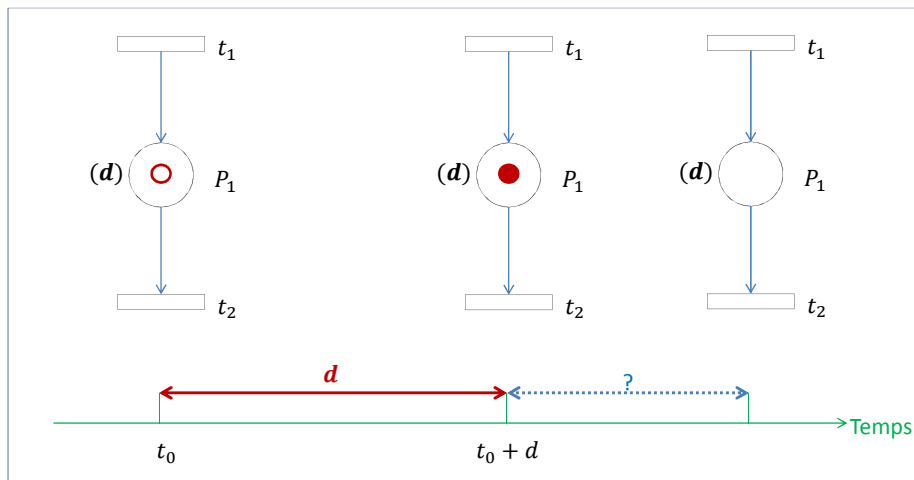


Figure 75 P-temporisation

Le franchissement des transitions se fait à vitesse maximale. Cela veut dire que dès qu'une transition est franchissable, elle est franchie. Cela peut parfois générer des temps d'attentes. La figure suivante illustre cette exécution à vitesse maximale.

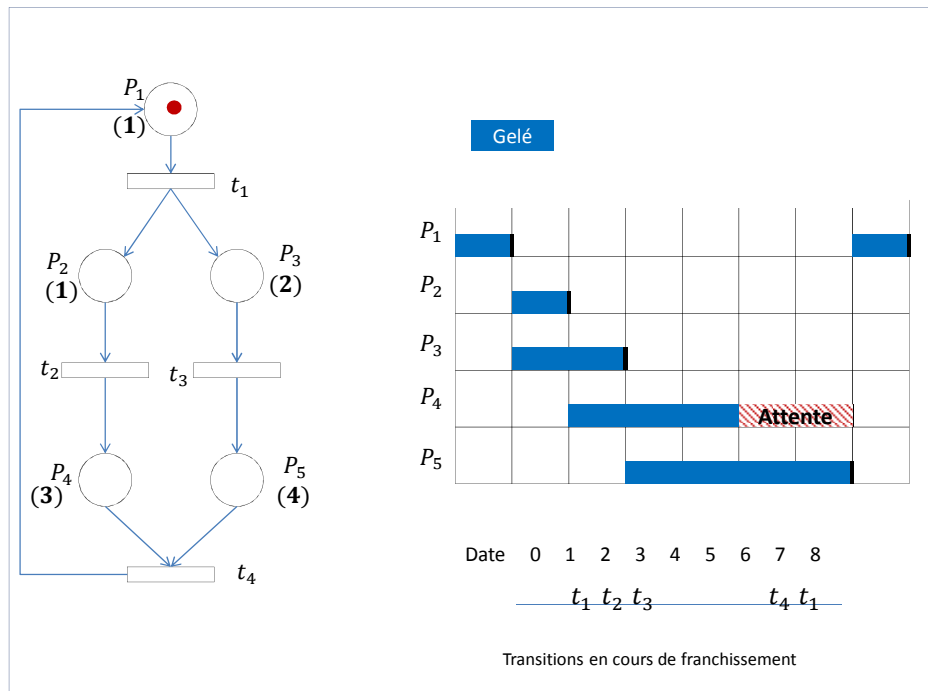


Figure 76 Exécution à vitesse maximale

On peut maintenant dessiner le graphe des marquages à vitesse maximale de ce réseau P-temporisé. Un arc entre deux marquages est étiqueté par T_k/dk , où T_k est la transition franchie et dk la durée écoulée entre les deux marquages.

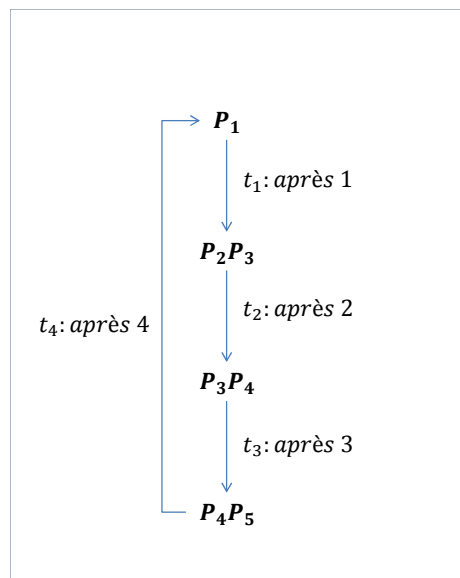


Figure 77 Graphe des marquages à vitesse maximale

On peut calculer la fréquence de franchissement F_j d'une transition T_j . Elle représente le nombre moyen de franchissements d'une transition par unité de temps, lorsque le régime devient périodique. C'est le cas pour un réseau borné fonctionnant à vitesse maximale.

A la figure suivante, le nombre moyen de marques dans la place P_1 est $d_1.F_2$ car les marques entrent à une fréquence de F_2 et un reste un temps d_1 . De même, dans P_2 , il y a en moyenne $d_2.F_1$ marques. On a : $d_1.F_2 + d_2.F_1 = M_0(P_1) + M_0(P_2)$.

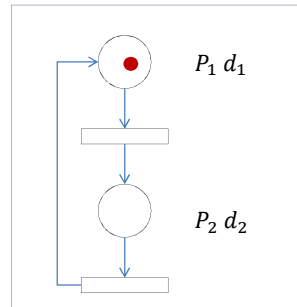


Figure 78 Calcul fréquence de franchissement

On observe que $F_1 = F_2$ et $d_1 = 2$ et $d_2 = 3$. On peut en déduire $F_1 = F_2 = 1/5$

Un second mode de fonctionnement est le fonctionnement **à vitesse propre**. Dans ce type de fonctionnement, un jeton ne reste dans une place qu'un temps égal à la durée associée à cette place. Ce type de fonctionnement ne s'applique pas à tous les réseaux P-temporisés. C'est un cas particulier du fonctionnement à vitesse maximale dans lequel il n'y a pas de temps d'attente. Le réseau de la figure 76 ne pourra pas fonctionner à vitesse propre. La figure suivante donne deux réseaux.

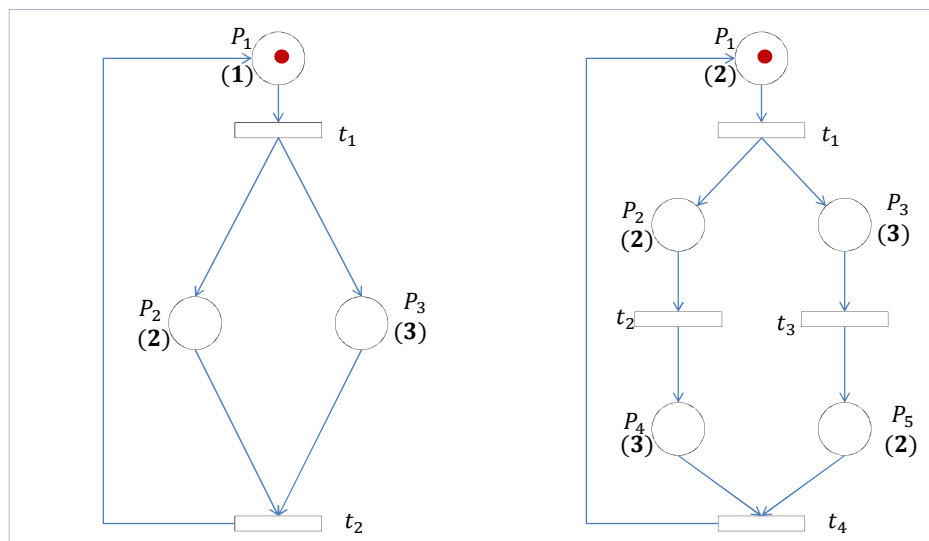


Figure 79 Fonctionnement à vitesse propre

Le réseau de gauche ne peut pas fonctionner en vitesse propre, car le tir de t_1 dépose un jeton dans les deux places de sortie. Il faut ensuite attendre que le jeton de P_3 soit dégelé pour tirer t_2 . Cela implique que le jeton reste 3 unités de temps dans P_2 alors que $d(P_2)=2$.

Par contre, le réseau de droite admet un fonctionnement à vitesse propre, qui est le fonctionnement à vitesse maximale. Car chaque jeton attend, dans sa place, la fin de la durée déterminée.

5.2.2 Les réseaux T-temporisés

Cette temporisation consiste à associer une durée non pas aux places, mais aux transitions. Le tir d'une transition se décompose alors en deux temps : **L'enclenchement** du tir et la **fin** du tir. C'est la durée spécifiée par la fonction de temporisation, qui sépare ces deux ci.

La figure suivante illustre les réseaux T-temporisés. On peut voir qu'il s'écoule exactement $d(t)$ entre le début et la fin du tir de la transition t . Pendant cet intervalle de temps, les jetons entrants sont **réservés**. Ils ne peuvent plus être utilisés pour valider d'autres transitions. A la fin du tir, ils disparaissent et les jetons produits par t_2 apparaissent, alors, dans un état **non réservé**.

En d'autres termes, au moment où le franchissement d'une transition validée est décidé, les jetons nécessaires sont réservés.

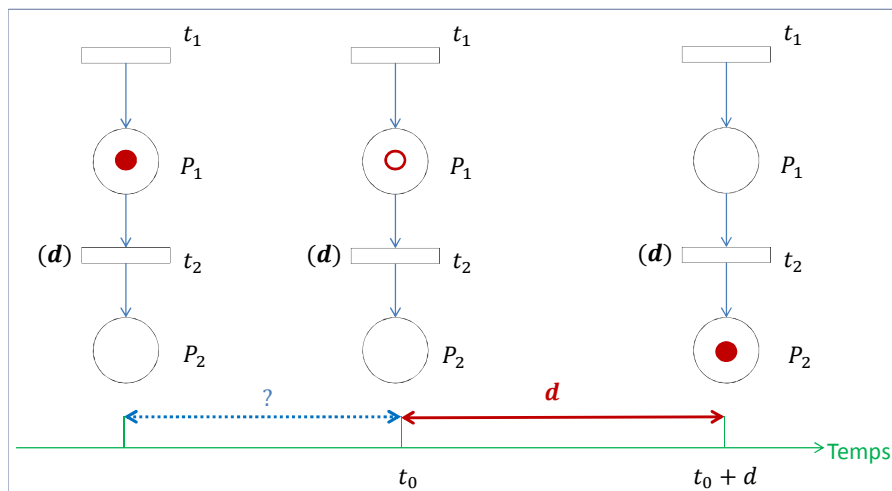


Figure 80 T-Temporisation

La différence avec les réseaux P-temporisés réside dans le fait que c'est la transition qui réserve le jeton et non la place.

Définition : Un réseau de Petri T-temporisé est défini par le doublet $\langle N, d \rangle$ où :

- N est un réseau places/transitions
- $d: T \rightarrow \mathbb{N}$ est la fonction de temporisation

Pour ces réseaux, le franchissement à vitesse maximale est aussi applicable. La figure suivante illustre ce type d'exécution.

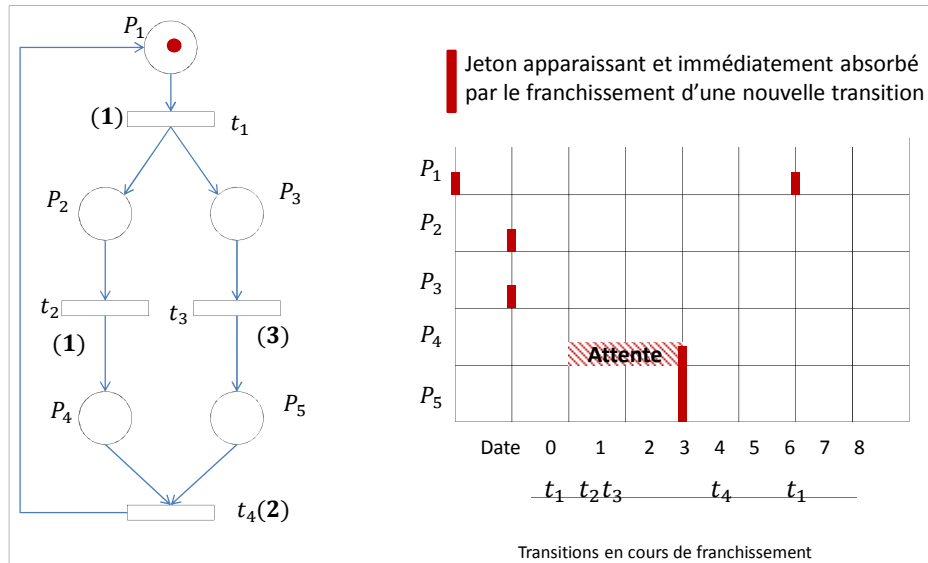


Figure 81 Exécution à vitesse maximale dans réseau T-temporisé

La réalisation d'un graphe des marquages de vitesse maximale est assez similaire des réseaux P-temporisés. La figure suivante illustre cette réalisation.

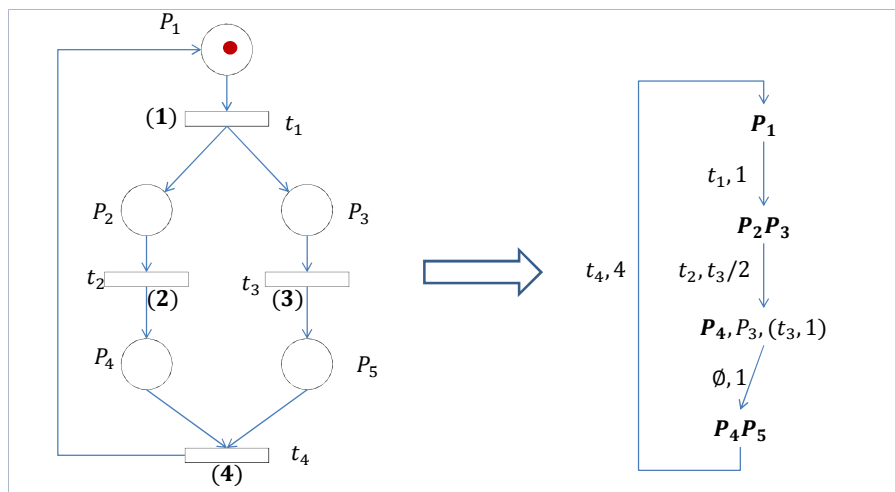


Figure 82 Réseau T-temporisé et son graphe des marquages à vitesse maximale

Le fonctionnement à vitesse propre est aussi applicable dans ces réseaux. La figure suivante nous donne une illustration d'un réseau et de son graphe des marquages à vitesse propre.

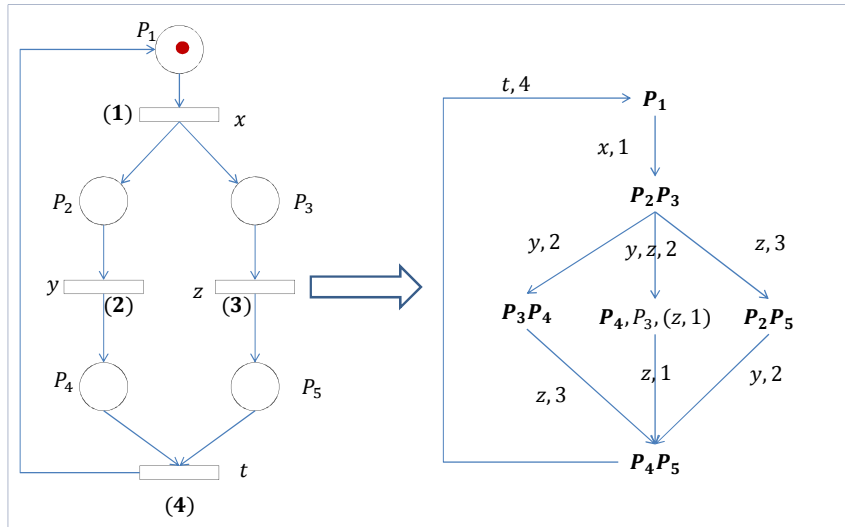


Figure 83 Réseau T-temporisé et son graphe des marquages à vitesse propre

5.2.3 Équivalence entre les deux temporisations

Les réseaux P-temporisés et les réseaux T-temporisés sont équivalents, car on peut passer de l'un à l'autre sans perte d'informations.

La figure suivante illustre le passage d'un réseau P-temporel à un réseau T-temporel. Il suffit de dupliquer chaque place P en P_{in} et P_{out} . Les transitions initialement en entrée de P sont les transitions entrées de P_{in} et P_{out} . Les temporisations des transitions initiales sont toutes nulles et $d'(t_{int})$ est égale à $d(P)$. On a pu insérer le tir de t_{int} entre celui de t_1 et celui de t_2 et ce tir durera d_1 unités de temps.

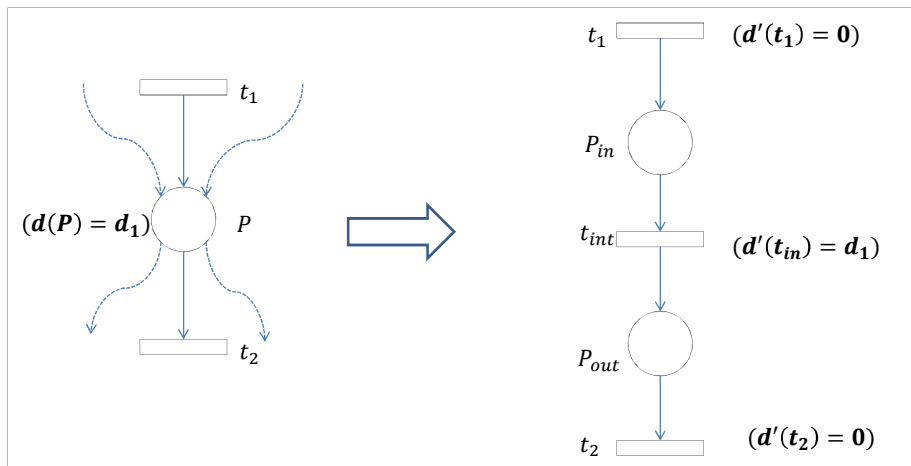


Figure 84 Le passage du P vers T-temporisation

La figure suivante illustre le passage d'un réseau T-temporel à un réseau P-temporel. Pour réaliser cette transformation, on décompose chaque transition t de temporisation $d(t)$ en deux transitions t_{deb} et t_{fin} et on introduit une place P_{int} de temporisation $d'(P_{int}) = d(t)$

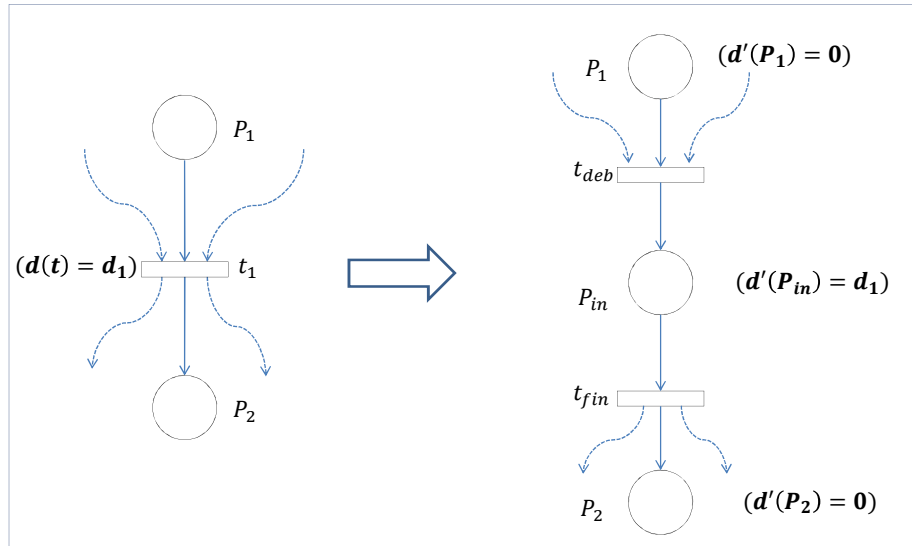


Figure 85 Le passage du T vers P-temporisation

5.3 Caractéristiques

Un réseau temporisé borné fonctionnant à vitesse maximale a toujours un comportement périodique au bout d'un temps infini.

Les propriétés des réseaux temporisés se vérifient de la même manière que les réseaux de Petri ordinaires. Cependant, contrairement aux réseaux colorés, les réseaux ordinaires et les réseaux temporisés ne sont pas équivalents. Les réseaux temporisés rajoutent une puissance d'expressivité supplémentaire qu'il est impossible d'introduire dans les réseaux ordinaires.

CHAPITRE 6 : Les réseaux temporels

6.1 Propriété manquante aux réseaux de Petri

Les réseaux temporels ont quelque chose en commun avec les réseaux temporisés : le temps. Cependant, l'objectif des réseaux temporels n'est pas de définir le temps d'une action, mais de pouvoir localiser les actions dans le temps. C'est assez délicat de déterminer le début et la fin précise d'une action, ce qui implique que celle-ci est placée dans une **plage horaire**.

Afin de mieux comprendre, voici un exemple de la vie courante: un conducteur se gare entre 8h et 9h du matin à un endroit convenu avec son collègue. Il attend son collègue pendant 30 minutes. Si au bout de ce temps, le collègue n'est pas arrivé, il repart. Si son collègue arrive, il repart aussitôt avec lui. Le collègue passe à cet endroit entre 8h30 et 9h45. Il attend la voiture au maximum 15 minutes. Et si au bout de cette durée, la voiture n'est pas là, il part de son côté. La figure suivante nous donne le réseau du système décrit.

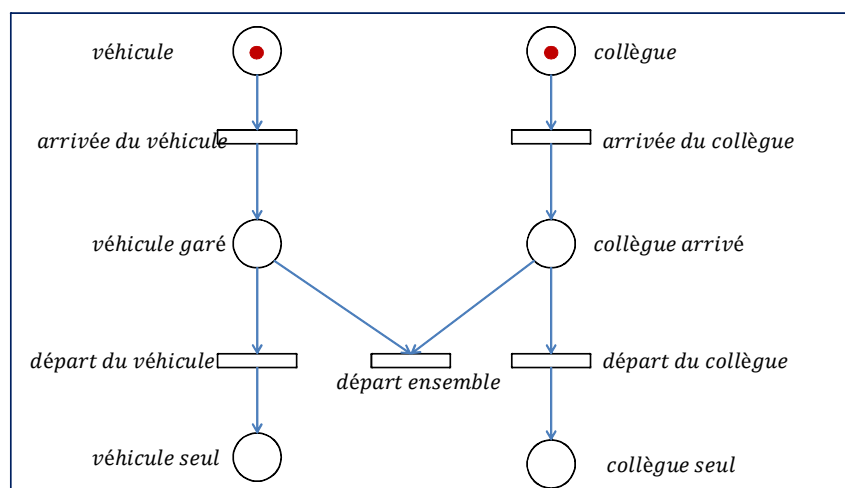


Figure 86 Réseau de Petri du covoiturage

Cependant, ce réseau n'est pas complet. Il ne tient pas compte des plages horaires données dans l'exemple. De plus, lorsque le jeton arrive dans la place « véhicule garé » et « collègue arrivé », aucune information ne précise la transition à tirer. Le réseau ne respecte pas les spécifications données.

On intègre, dans le réseau, les intervalles de temps qui rendent les jetons disponibles ou indisponibles. Un jeton ne sera disponible que pendant l'intervalle de temps donné : la transition est franchissable durant cet intervalle.

6.2 Définition

Le principe général des réseaux temporels consiste à associer une **durée de sensibilisation** à chaque transition. Celle-ci n'est franchissable que si le marquage la rend valide et uniquement pendant sa durée de sensibilisation $\theta_s(t)$. La différence avec les réseaux temporisés est que les jetons ne sont pas gelés ou bloqués dans les places. Ils doivent uniquement attendre qu'une transition devienne valide, grâce à la durée de sensibilisation, pour pouvoir la franchir. Cela

n'empêche pas un jeton d'être tiré par une autre transition qui se valide plus rapidement. Le domaine d'application de ce modèle, la validation des protocoles de communication, a fait que dès le départ, il a été défini une durée imprécise de sensibilisation de la transition t sous la forme d'un intervalle $i(t) = [\theta_{s,min}(t), \theta_{s,max}(t)]$. Dans la figure suivante, la transition est franchissable entre le temps 3 et le temps 7. Le jeton a 4 unités de temps pour franchir cette transition.

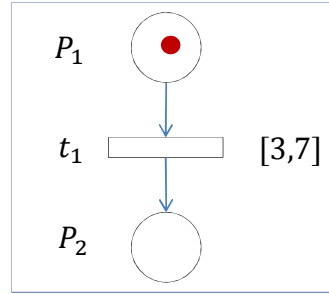


Figure 87 Illustration d'un réseau temporel

Il existe, comme pour les réseaux de Petri temporisés, les réseaux de Petri T-temporels et les réseaux de Petri P-temporels. Nous nous intéresserons uniquement, par manque de temps, aux réseaux de Petri T-temporels

Définition : Un réseau de Petri T-temporel est un couple (N, IS) où :

- N est un réseau places/transitions muni d'un marquage initial M_0
- IS est la fonction qui à chaque transition t fait correspondre un intervalle fermé rationnel $i(t) = [\theta_{s,min}(t), \theta_{s,max}(t)]$ qui décrit une durée de sensibilisation. On appelle aussi IS la fonction **intervalle de tir statique** où :
 - $\theta_{s,min}(t)$ est la **date statique de tir au plus tôt**
 - $\theta_{s,max}(t)$ est la **date statique de tir au plus tard**.

La règle de franchissement est alors la suivante : une transition est tirée entre le temps $\theta_{s,min}$ et $\theta_{s,max}$.

De manière plus formelle,

La règle de tir d'une transition t à une date θ est possible si :

- t est sensibilisée par le marquage M , c'est-à-dire $M \geq \text{Pré}(t)$
- θ n'est pas inférieure à la date de tir au plus tôt de t : $\theta \geq \theta_{s,min}(t)$
- θ n'est pas supérieure à la date de tir au plus tard de t : $\theta \leq \theta_{s,max}(t)$
- θ n'est pas supérieure à la date de tir au plus tard d'aucune transition sensibilisée par le marquage M : $\forall k / M \geq \text{Pré}(k) \Rightarrow \theta \leq \theta_{s,max}(k)$

Cela veut dire que la transition pourra être tirée si sa date est supérieure ou égale à $\theta_{s,min}$. Et s'il n'existe aucune autre transition sensibilisée, dont l'intervalle de tir expire avant la date. Dans ce cas, cette transition devra être tirée ou invalidée par le tir d'une autre transition avant la date.

La règle de tir est illustrée à la figure suivante.

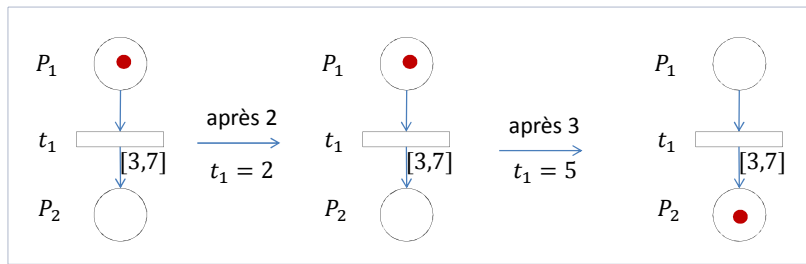


Figure 88 Règle de tir dans un réseau temporel

Les figures suivantes donnent le réseau temporel et son fonctionnement dans l'exemple du covoiturage.

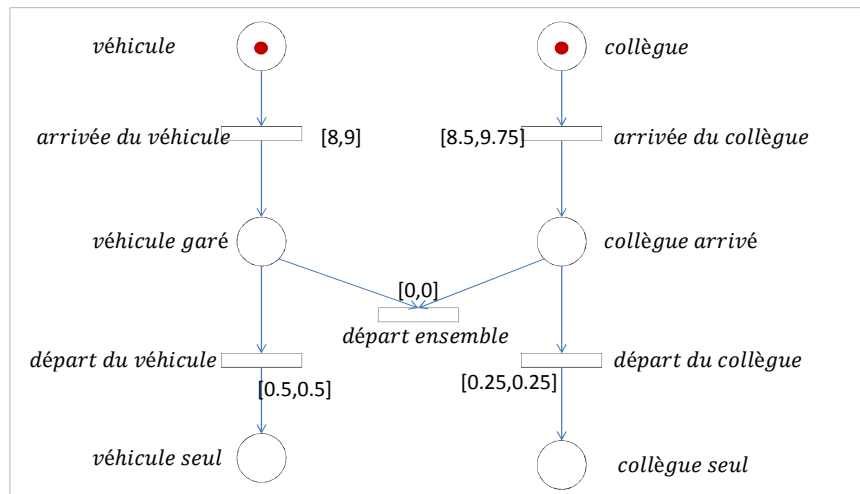


Figure 89 Réseau temporel à l'instant t=0

Le véhicule arrive à 8h15. La transition « arrivée du véhicule » est tirée. Un jeton (rouge) est placé dans la place « Véhicule garé ». Attention, cela ne valide pas la transition « départ ensemble » mais ce marquage sensibilise la transition « départ du véhicule ». Si le collègue arrive à 8h35, un jeton (jaune) est mis dans la place « collègue arrivée ». La transition « départ ensemble » est tirée immédiatement car le marquage la sensibilise et la durée est comprise dans l'intervalle.

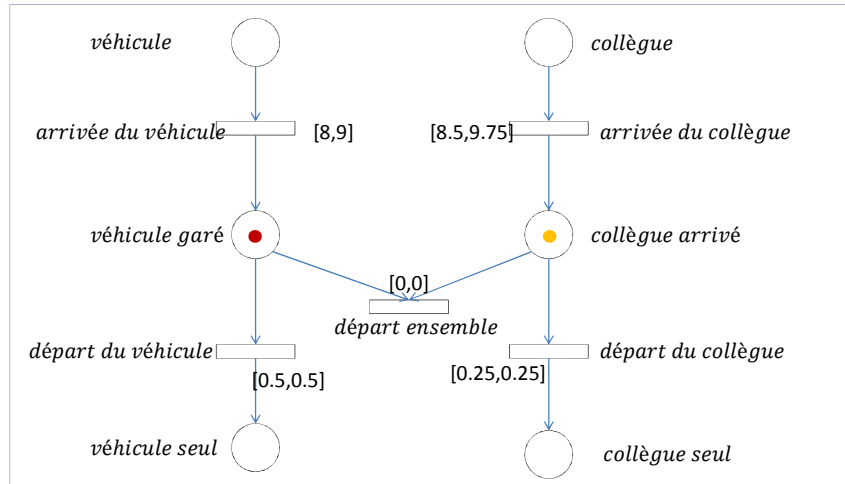


Figure 90 Réseau temporel à 8h15(rouge) et 8h35(jaune)

Par contre, à la figure 91. Le véhicule est arrivé à 8h15 mais le collègue n'arrive lui que à 9h. Il arrive trop tard. En effet, la transition « départ du véhicule » (jeton bleu) est tirée car l'échéance a été dépassée à 8h45. Le collègue va attendre 15 minutes sur place et repartira seul (jeton vert).

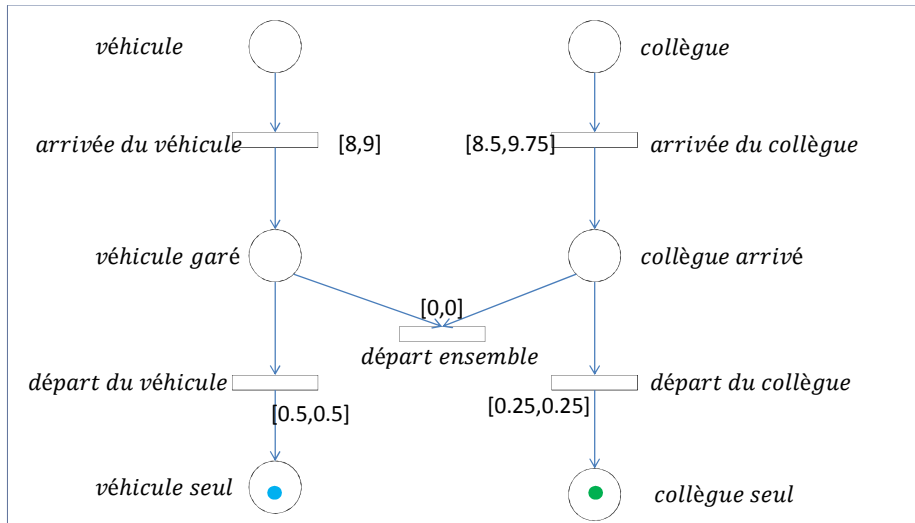


Figure 91 Réseau temporel à 8h45 (bleu) et à 9h15(vert)

La suite de cette section nous donne la description du fonctionnement d'un réseau temporel.

6.2.1 État d'un réseau temporel et tir de transitions

Nous définissons, en premier lieu, la notion d'état. La description de l'état d'un réseau temporel doit comprendre : des informations liées à l'état du système (le marquage) et des informations temporelles (intervalles de tir des transitions sensibilisées).

Définition : L'état d'un réseau de Petri temporel est un couple (M, I) où :

- M est le marquage
- I est le vecteur d'intervalles, de taille $|T|$ dont la $i^{\text{ème}}$ composante est égale à :
 - \emptyset si la transition t_i n'est pas sensibilisée
 - L'intervalle dynamique de tir de t_i sinon

Notons que I peut être interprété comme une application intervalle de tir. Considérons le réseau de la figure suivante.

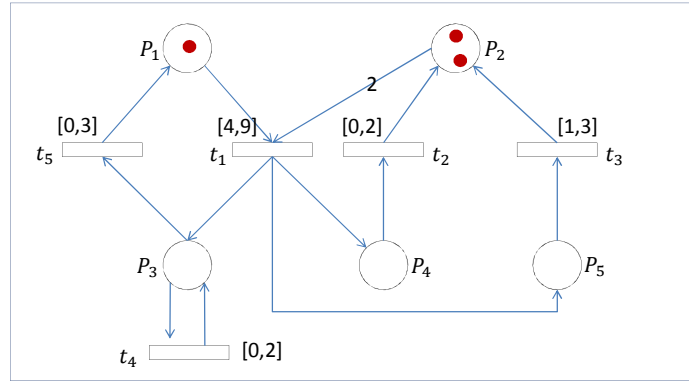


Figure 92 Illustration des états d'un réseau temporel

Les états peuvent être représentés par des paires (M, D) , dans M est un marquage et D un ensemble de vecteurs de dates appelé domaine de tir. L' i -ième projection de l'espace D est l'intervalle de tir $I(t_i)$ associé à la i -ième transition sensibilisée. Les domaines de tir sont décrits par des systèmes d'inéquation linéaires avec une variable par transition sensibilisée.

- L'état initial $E_0 = (M_0, D_0)$ représenté par :
 $M_0: P_1, P_2 * 2$
 $D_0: 4 \leq t_1 \leq 9$
- Le tir de t_1 depuis E_0 , à une date relative $\theta_1 \in [4, 9]$, mène à $E_1 = (M_1, D_1)$.
 $M_1: P_3, P_4, P_5$
 $D_1: 0 \leq t_2 \leq 2$
 $1 \leq t_3 \leq 3$
 $0 \leq t_4 \leq 2$
 $0 \leq t_5 \leq 3$
- Le tir de t_2 depuis E_1 , à une date relative $\theta_2 \in [0, 2]$, mène à $E_2 = (M_2, D_2)$.
 $M_2: P_2, P_3, P_5$
 $D_2: \max(0, 1 - \theta_2) \leq t_3 \leq 3 - \theta_2$
 $0 \leq t_4 \leq 2 - \theta_2$
 $0 \leq t_5 \leq 3 - \theta_2$
- Comme θ_2 peut prendre toute valeur réelle dans $[0, 2]$, l'état E_1 admet une infinité de successeurs par t_2

Les définitions de la franchissabilité et du tir de transition doivent aussi intégrer la prise en compte des contraintes temporelles.

Définition : Le tir d'une transition t à la date θ depuis un état $E = (M, I)$ conduit à l'état $E' = (M', I')$:

- M' est le nouveau marquage : $M' = M - \text{Pré}(t) + \text{Post}(t)$
- $I'(k)$ est défini $\forall k$ comme suit :
 - Si k non sensibilisée par M' , alors $I'(k) = \emptyset$
 - Si $k \neq t$, est sensibilisé par M' , et n'est pas en conflit avec t pour M alors :

- $I'(k) = \begin{cases} [\max(0, \theta_{s,\min}(k) - \theta), \theta_{s,\max}(k) - \theta] \\ [\max(0, \theta_{s,\min}(k) - \theta), \infty] \end{cases}$
- Sinon $I'(k) = IS(k)$

La règle de tir ci-dessus définit une relation d'accessibilité sur l'ensemble des états d'un réseau temporel. Les séquences de tirs sont définies comme pour les réseaux de Petri ordinaires, un échéancier de tir associe une séquence de tirs s à une séquence u de dates. Un échéancier (s, u) est dit réalisable depuis un état E , aux dates relatives de tir qui leur correspondent dans la séquence u . Le fonctionnement d'un réseau temporel peut être caractérisé par l'ensemble des états accessibles depuis son état initial ou de façon duale, par l'ensemble des échéanciers réalisables depuis son état initial.

Construire le graphe d'accessibilité des états d'un réseau temporel est en général impossible : les transitions pouvant être tirées à tout instant dans leurs intervalles de tir. Les états admettent en général une infinité de successeurs. Les **Classes d'États** vont être introduites pour fournir une représentation finie d'un nouveau graphe : **le graphe des classes**.

6.2.2 Classes d'états et graphe des classes

L'ensemble des états d'un réseau temporel est en général **infini** pour deux raisons :

1. Un état peut admettre une infinité de successeurs
2. Un réseau peut admettre des échéanciers passant par des états tous différents.

La solution est de regrouper en classes d'états, les états obtenus depuis l'état initial par le tir d'une même séquence. Tous ces états ont dès lors le même marquage. En considérant l'ensemble de tous les états que l'on peut atteindre à partir de l'état initial par le tir d'échéanciers ayant pour support la même séquence. Cet ensemble obtenu définit la **Classe d'États** associée à la séquence de tir. Elles peuvent être représentées par un marquage et un domaine de tir, comme l'étaient les états. Le marquage est celui des états agglomérés dans la classe, le domaine de tir de la classe est la réunion des domaines de tir des états constituant la classe.

Définition : Une classe d'état $C = (M, D)$ est un regroupement d'états tel que :

- Tous les états ont le même marquage M ,
- D est une réunion de tous les domaines de tir des états de C

Reprenons le schéma de la figure 92 pour construire quelques les classes du réseau.

La classe initiale C_0 est définie comme l'état initial E_0 . Le tir de t_1 depuis C_0 conduit à une classe C_1 identique à l'état E_1 . Le tir de t_2 depuis C_1 conduit à la classe $C_2 = (M_2, D_2)$, avec $M_2 = (p_2, p_3, p_5)$ et D_2 calculé en 4 étapes :

- a. $D_2(a)$ est obtenu en ajoutant à D_1 , les conditions de tirabilité de t_2

$$\begin{aligned} 0 &\leq t_2 \leq 2 \\ 1 &\leq t_3 \leq 3 \\ 0 &\leq t_4 \leq 2 \\ 0 &\leq t_5 \leq 3 \\ t_2 &\leq t_3 \\ t_2 &\leq t_4 \\ t_2 &\leq t_5 \end{aligned}$$

b. On cherche les variables associées aux transitions en conflit avec t pour les éliminer.
Ici, aucune transition n'est en conflit avec t_2 , on a $D_2(b) = D_2(a)$

c. Le changement d'origine produit le système suivant :

$$\begin{array}{lll} 0 \leq t_2 \leq 2 & 1 \leq t_2 + t_3 \leq 3 & t_2 \leq t_2 + t_3 \\ & 0 \leq t_2 + t_4 \leq 2 & t_2 \leq t_2 + t_4 \\ & 0 \leq t_2 + t_5 \leq 3 & t_2 \leq t_2 + t_5 \end{array}$$

Depuis lequel $D_2(c)$ est obtenu par élimination de t_2

$$\begin{array}{ll} 0 \leq t_3 \leq 3 & t_4 - t_3 \leq 1 \\ 0 \leq t_4 \leq 2 & t_5 - t_3 \leq 2 \\ 0 \leq t_5 \leq 3 & \end{array}$$

d. On recherche les nouvelles transitions sensibilisées et on ajoute les contraintes. Il n'y a aucune transition, on a donc $D_2 = D_2(c)$

Pour ce réseau, il y a exactement 12 classes. Nous en avons trouvé 3. La recherche des autres classes est à votre convenance.

L'étape suivante consiste à construire le graphe des classes, en suivant la même démarche que la construction du graphe des marquages pour les réseaux de Petri ordinaires.

La règle de franchissabilité s'exprime par $C = (M, D)(t > \text{ si et seulement si}$

1. $M(t >$
2. $\exists d \in D$ tel que $\forall t'$ sensibilisée, $d(t) \leq d(t')$

La figure suivante nous donne un réseau temporel avec son graphe des classes. Le tableau ci-dessous est l'ensemble des classes d'états.

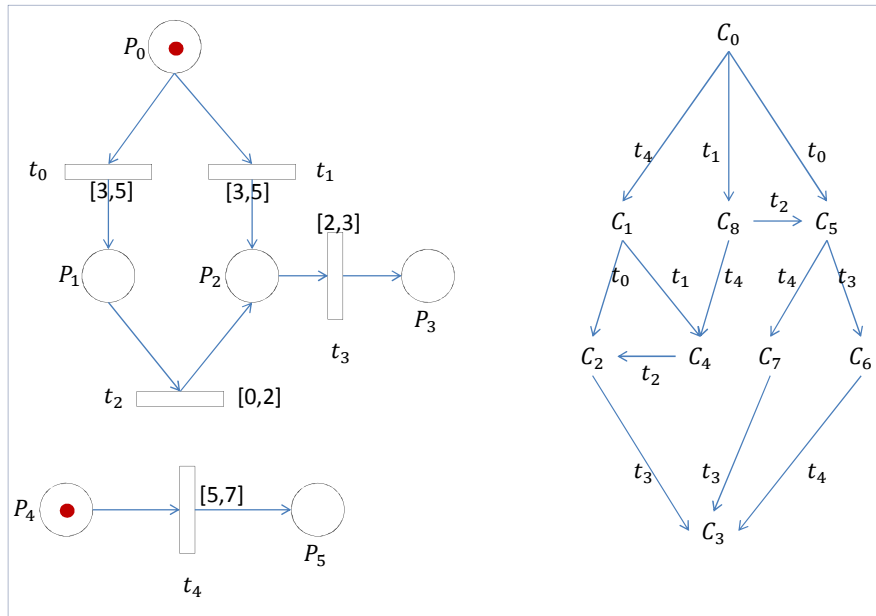


Figure 93 Réseau temporel et son graphe des classes

C	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
M	P_0, P_4	P_0, P_5	P_2, P_5	P_3, P_5	P_1, P_5	P_2, P_4	P_3, P_4	P_2, P_5	P_1, P_4
D	$5 \leq t_4 \leq 7$ $3 \leq t_0 \leq 5$ $3 \leq t_1 \leq 5$	$0 \leq t_0 \leq 0$ $0 \leq t_1 \leq 0$	$2 \leq t_3 \leq 3$		$0 \leq t_2 \leq 2$	$2 \leq t_3 \leq 3$ $0 \leq t_4 \leq 4$	$0 \leq t_4 \leq 2$	$0 \leq t_3 \leq 3$	$0 \leq t_4 \leq 4$ $0 \leq t_2 \leq 2$

6.3 Caractéristiques

La méthode d'analyse exposée dans cet article permet pour les réseaux temporels, une analyse d'accessibilité semblable à celle permise pour les réseaux de Petri par la technique du graphe des marquages. Cette technique a été utilisée dans de nombreux travaux universitaires ou industriels. Elle a été intégrée à plusieurs outils d'analyse de systèmes. Les limites intrinsèques de la méthode ne doivent toutefois pas être perdues de vue. Une première limite est qu'on ne peut énoncer de condition nécessaire et suffisante pour la propriété bornée des réseaux temporels. Une autre limite est que le nombre de classes d'états d'un réseau temporel peut être très grand.

Rappelons cependant qu'un réseau de Petri est borné si le marquage de toute place admet une borne supérieure. La propriété bornée est indécidable pour les réseaux temporels, mais l'ensemble des domaines de tir d'un réseau temporel est toujours fini. Le graphe des classes d'un réseau temporel est donc fini si et seulement si ce réseau est borné

Ainsi, toute condition suffisante pour la propriété bornée fournira une condition suffisante pour la propriété de finitude du graphe des classes.

Propriété : Un réseau temporel est borné s'il n'admet pas de paires de classes d'états marqués. Si classes $C = (M, D)$ et $C' = (M', D')$ telles que :

1. C' est accessible depuis C
2. $M' \geq M$
3. $D=D'$
4. $\forall p. M'(p) > M(p) \Rightarrow M'(p) > \text{Max}\{\text{Pré}(p, t), t \in T\}$

Les propriétés 1 et 4 sont nécessaires pour qu'un réseau soit non borné, mais pas suffisantes. Cette propriété permet, par exemple, de démontrer que les réseaux de la figure suivante *a* et *b* sont bornés. Mais il ne permet pas de démontrer que le réseau *c* est borné, bien que celui-ci n'admette que 48 classes d'états.

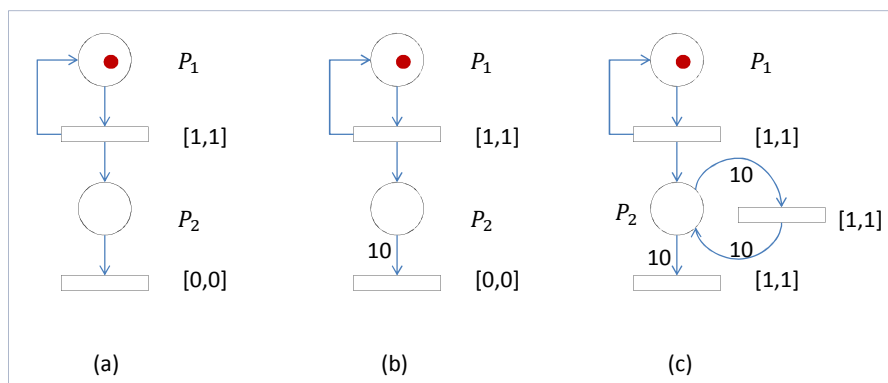


Figure 94 Trois réseaux temporels bornés

Pour les réseaux que l'on ne peut montrer borné à l'aide de la propriété, il reste possible de procéder à une énumération contrainte. Par exemple en bornant arbitrairement le nombre de classes à énumérer ou le marquage de chaque place. L'analyse structurelle fournit aussi des conditions suffisantes pour le caractère borné.

CHAPITRE 7 : Les réseaux stochastiques

7.1 Propriété manquante aux réseaux de Petri

Avant d'introduire la propriété manquante aux réseaux de Petri, nous allons expliquer le principe d'un processus stochastique et la manière dont on peut l'analyser. Sans ces explications, il sera compliqué de comprendre concrètement l'intérêt des réseaux stochastiques.

7.1.1 Processus stochastique et analyse markovienne

Un **processus stochastique** est une famille $\{X_t\}_{t \in T}$ de variables aléatoires indexées par le temps t . Les mots *processus* et *stochastique* signifient respectivement *fonction* et *aléatoire*. Alors qu'une variable aléatoire X associe à chaque $\omega \in \Omega$ une réalisation $X(\omega)$, un processus stochastique $\{X_t\}_{t \in T}$ associe à chaque ω une fonction (ou trajectoire) $\{X_t(\omega)\}_{t \in T}$:

$$\begin{aligned} T &\rightarrow E \\ t &\mapsto X_t(\omega) \end{aligned}$$

Où E est l'espace d'arrivée des variables aléatoires X_t . Passer des variables aléatoires aux processus stochastiques revient à passer en analyse des points de fonctions. À titre d'exemple, la trajectoire d'une mouche en fonction du temps peut être modélisée par un processus stochastique à valeurs dans $E = \mathbb{R}^3$. Lorsque l'ensemble des temps T est au plus dénombrable (par exemple $T = \mathbb{N}$), on parle de processus stochastiques à temps discret. Lorsqu'il est continu ($T = [0; t_0]$ ou $T = \mathbb{R}_+$), on parle de processus stochastique à temps continu.

Un processus stochastique est **Markovien** si, conditionnellement à sa valeur présente au temps t , son évolution future est indépendante de son passé. En d'autres termes, en connaissant le présent, la prédiction du futur ne pourra pas être précisée par des informations du passé. Il n'y a que l'état présent qui soit utile pour l'état futur. Un modèle d'évolution dynamique en temps discret dans lequel on fait dépendre l'évolution future de l'état présent et du hasard est une **chaîne de Markov**. C'est un processus stochastique à temps discret.

Définition : Soit $(X_n)_{n \in \mathbb{N}}$ une suite de variables aléatoires à valeurs dans un espace E fini ou dénombrable, appelé *espace d'états*. On dit que Soit $(X_n)_{n \in \mathbb{N}}$ est une chaîne de Markov si

$$\mathbb{P}(X_{n+1} = j | X_n = i, X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0) = \mathbb{P}(X_{n+1} = j | X_n = i),$$

Pour tout entier $n \in \mathbb{N}$, pour tout état j et pour toute suite d'états $i_0, i_1, \dots, i_{n-1}, i$ pour lesquels la probabilité conditionnelle a un sens, c'est-à-dire

$$\mathbb{P}(X_n = i, X_{n-1} = i_{n-1}, \dots, X_1 = i_1, X_0 = i_0) > 0$$

Il faut comprendre une chaîne de Markov $(X_n)_{n \in \mathbb{N}}$ comme une promenade dans l'espace d'états E , la variable X_n indiquant l'état dans lequel on est à l'instant n . La variable aléatoire X_0 représente l'état initial duquel démarre la chaîne. Selon le contexte, X_0 pourra être aléatoire ou déterministe. La propriété de Markov signifie que, connaissant le dernier état visité n , la loi du prochain état visité ne dépend pas des états visités depuis l'instant 0 jusqu'à l'instant $n - 1$. Plus prosaïquement : conditionnellement au présent, le futur ne dépend pas du passé, mais du présent.

7.1.2 Adaptation des réseaux temporisés

L'utilisation des réseaux de Petri temporisés permet de construire des modèles réalistes pour évaluer des systèmes à événements discrets par simulation. En effet, rien n'empêche de modifier la fonction θ_f pour associer aux transitions non pas une valeur constante, mais le résultat d'un tirage aléatoire. Ces réseaux temporisés ont servi à vérifier formellement un certain nombre de protocoles.

Néanmoins la taille du graphe de couverture explose très vite lorsque les intervalles de temps sont trop divers. Le problème principal vient du fait que la description de l'état de ces réseaux doit comprendre non seulement le marquage, mais aussi les informations temporelles. Afin de pouvoir utiliser la puissance de l'analyse Markovienne, il faut que les systèmes soient sans mémoire du passé, c'est-à-dire que si un événement produit un franchissement de transition t et transforme le marquage M_1 en M_2 , l'évolution future des transitions qui étaient sensibilisées par M_1 avant le franchissement de t doit être identique à celle qu'elles subiraient si elles venaient juste d'être sensibilisées par M_2 .

Les réseaux stochastiques sont donc des réseaux de Petri où les durées de sensibilisation associées aux transitions sont définies par des distributions géométriques⁵ et exponentielles⁶ afin de pouvoir construire un processus Markovien équivalent et ainsi analyser les comportements du réseau.

7.1.3 Durée de sensibilisation stochastique

Au lieu de prendre de façon égale toutes les valeurs de l'intervalle $([\theta_{min}, \theta_{max}])$ comme dans les réseaux de Petri temporels, on peut supposer que la durée de sensibilisation est une variable stochastique θ avec une distribution de probabilité exponentielle :

$$\Pr[\theta \leq t] = 1 - e^{-\lambda t}$$

La fonction $\Pr[\theta \leq t]$ décrit la probabilité pour que le franchissement ait lieu avant t , c'est-à-dire pour que la durée de sensibilisation soit inférieure à t . On en déduit que la valeur moyenne de la durée de sensibilisation est

$$\bar{\theta} = \int_0^{\infty} (1 - \Pr[\theta \leq t]) dt = \int_0^{\infty} e^{-\lambda t} dt = \frac{1}{\lambda}$$

Où λ est appelé taux de transition.

⁵ La distribution géométrique est une distribution de probabilité discrète. Elle permet de modéliser « le nombre de fois qu'il faut répéter l'épreuve pour obtenir le premier succès »

⁶ La distribution exponentielle est une distribution de probabilité continue. Elle s'intéresse à la variable aléatoire qui représente le temps d'attente pour la réalisation d'un événement ou le temps d'attente entre la réalisation de deux événements successifs.

7.2 Définition

Un réseau de Petri stochastique est un réseau de Petri temporisé doté d'une mesure de probabilité sur l'espace des trajectoires, c'est-à-dire que les séquences de franchissements sont mesurables en considérant un espace aléatoire.

Définition : Un réseau de Petri stochastique est un couple (N, Λ) où :

- N est un réseau de Petri avec un marquage initial
- Λ est une fonction qui à chaque transition t associe un taux de transition $\lambda_t = \Lambda(t)$.

Cela revient à associer à chaque transition un intervalle de sensibilisation continu $[0, \infty[$ avec une distribution exponentielle. Ceci explique que l'ensemble des marquages accessibles est le même que celui du réseau sous-jacent.

7.2.1 Obtention d'un processus de Markov

Soient deux marquages accessibles M_i et M_j tels qu'une seule transition t_k soit telle que $M_i \xrightarrow{t_k} M_j$ le taux de transition de l'état M_i vers l'état M_j est bien évidemment $\Lambda(t_k)$.

Si deux transitions t_k et t_m permettent de passer de M_i à M_j , alors le taux de transition de M_i vers M_j sera $\Lambda(t_k) + \Lambda(t_m)$.

À partir d'un réseau stochastique, on peut construire un processus de Markov. Les états seront l'ensemble des marquages accessibles du réseau. À partir de la fonction Λ on écrit directement la matrice Q des taux de transitions. La colonne j de Q décrit l'évolution de la probabilité de π_j du marquage M_j au cours du temps. Le terme q_{ij} est le taux de transition pour arriver dans l'état M_j à partir de l'état M_i . Le terme q_{ii} est négatif, car il décrit le taux de transition permettant de quitter l'état de M_j .

Illustrons tous ces nouveaux concepts par un exemple.

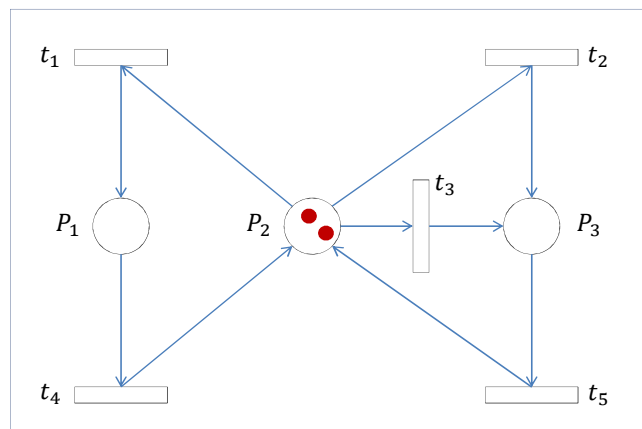


Figure 95 Réseau de Petri stochastique

Soit le réseau de Petri stochastique de la figure précédente. Supposons que les taux de transition soient les suivants : $\Delta(t_i) = \lambda_i$ pour $i = 1 \dots 4$ et : $\Delta(t_i) = M(P_3) \cdot \lambda_5$.

Le graphe des marquages est donné à la figure suivante :

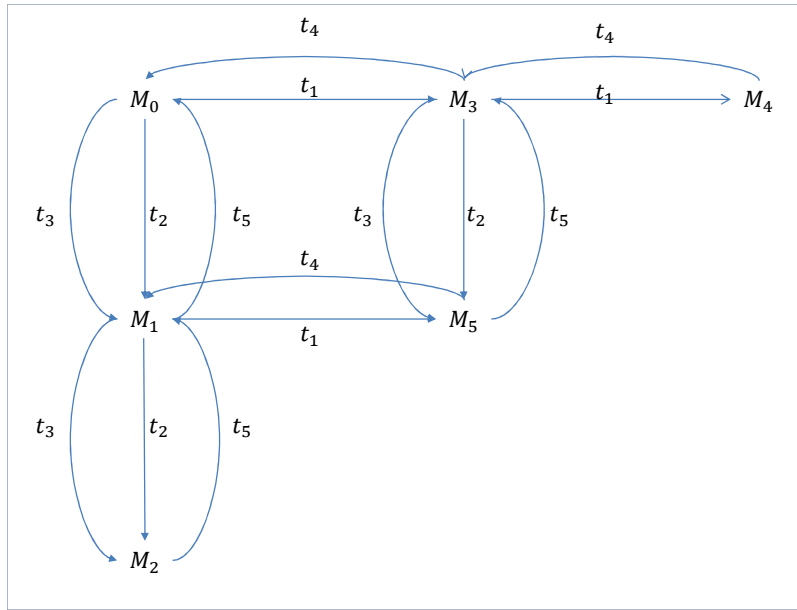


Figure 96 Graphe des marquages accessibles

La matrice des taux de transition est donnée par la matrice M avec les marquages suivants :

- $M_0 = P_2^2, M_1 = P_2 P_3, M_2 = P_3^2$
- $M_3 = P_1 P_2, M_4 = P_1^2, M_5 = P_1 P_3$

La matrice de transition est la suivante :

	M_0	M_1	M_2	M_3	M_4	M_5
M_0	$-\lambda_1 - \lambda_2 - \lambda_3$	$\lambda_2 + \lambda_3$	0	λ_1	0	0
M_1	λ_5	$-\lambda_1 - \lambda_2 - \lambda_3 - \lambda_5$	$\lambda_2 + \lambda_3$	0	0	λ_1
M_2	0	$2 \cdot \lambda_5$	$-2 \cdot \lambda_5$	0	0	0
M_3	λ_4	0	0	$-\lambda_1 - \lambda_2 - \lambda_3 - \lambda_4$	λ_1	$\lambda_2 + \lambda_3$
M_4	0	0	0	λ_4	$-\lambda_4$	0
M_5	0	λ_4	0	λ_5	0	$-\lambda_4 - \lambda_5$

On peut donc constater que :

- La première colonne de la matrice des taux de transition donne l'équation suivante (π_i est la probabilité du marquage M_i) :

$$\frac{d\pi_0}{dt} = -(\lambda_1 + \lambda_2 + \lambda_3) \cdot \pi_0 + \lambda_5 \cdot \pi_1 + \lambda_4 \cdot \pi_3$$

- La somme des éléments d'une ligne de Q est toujours égale à 0, car les arcs sortants d'un état M_i sont aussi les arcs entrants aux états M_j à partir de M_i

Conclusion

Les réseaux de Petri forment un puissant outil de modélisation, d'analyse et de simulation de systèmes. Ils permettent en outre une vérification formelle de ces systèmes, garantissant leur bon fonctionnement.

Cependant, les réseaux de Petri sont limités par leur propre conception. Des extensions ont dès lors été inventées pour adapter les réseaux à des modélisations plus exigeantes. Grâce à ces celles-ci, les réseaux de Petri peuvent modéliser des systèmes plus complexes avec des spécifications particulières. Cela n'aurait pu être réalisé par les réseaux de Petri ordinaires.

Dans ce document sont présentées dix extensions des réseaux de Petri. Chacune permet d'apporter une solution à une limite de conception initiale rencontrée lors de la modélisation d'un système particulier.

EXTENSIONS	UTILITÉ
LES RÉSEAUX À ARC INHIBITEUR	Permettent de modéliser le test à zéro
LES RÉSEAUX SYNCHRONISÉS	Permettent d'interagir avec des événements externes au système
LES RÉSEAUX PRÉDICATS/TRANSITIONS	Permettent d'introduire des variables et des opérations dans le réseau et les jetons. Approche semblable à celle des réseaux colorés
LES RÉSEAUX À FILES	Permettent de gérer l'ordre d'arrivée et la gestion des jetons. Ils ajoutent la notion de mémoire dans les réseaux de Petri
LES RÉSEAUX À OBJETS	Permettent de modéliser des systèmes qui souhaitent intégrer la notion d'objet dans la conception
LES RÉSEAUX FLOUS	Permettent, avec la logique floue, d'intégrer des règles floues. Ces réseaux se rapprochent de la réflexion humaine
LES RÉSEAUX COLORÉS	Permettent d'introduire une identification formelle des jetons et d'y insérer des informations
LES RÉSEAUX TEMPORISÉS	Permettent d'introduire des informations temporelles sur la durée des activités des systèmes
LES RÉSEAUX TEMPORELS	Permettent d'introduire des informations temporelles sur la durée ou le délai des opérations ou actions des systèmes
LES RÉSEAUX STOCHASTIQUES	Permettent d'intégrer des processus Markovien.

Les extensions de ce document ont été choisies selon les deux critères suivants.

Premièrement, nous avons sélectionné ces extensions pour montrer leur diversité. En effet, elles n'apportent pas toutes la même solution à une limite. Au contraire, elles répondent chacune à un problème particulier. Il est normalement possible de modéliser une plus grande série de limites et de spécifications avec les extensions.

Deuxièmement, les extensions ont été sélectionnées selon un critère de popularité. En effet, elles ont fait l'objet de nombreux travaux et d'applications concrètes. Cela justifie le choix de ces extensions car elles ont montré une plus grande utilisation que d'autres.

Cependant, ces extensions sont moins performantes au point de vue de la vérification formelle. En effet, certaines extensions ne permettent pas d'analyser les propriétés des réseaux et donc de les vérifier formellement. Or, cette analyse fait la richesse initiale des réseaux de Petri. Il y a donc un réel dilemme entre la capacité d'analyse des réseaux et un plus grand pouvoir d'expression de modélisation.

Enfin, il est bon de se demander si ces extensions constituent une réelle avancée pour les réseaux de Petri. Et si elles continueront d'être utilisées dans des applications concrètes. La complexité de leur apprentissage pourrait en effet inciter certains modélisateurs à s'orienter vers des outils de modélisation moins formels et ainsi d'augmenter le risque de réaliser des analyses incorrectes. Cependant, les réseaux de Petri avec ses extensions restent un outil puissant et utile pour la modélisation.

Bibliographie

CHAPITRE 1 : CONCEPTS ET NOTIONS PRÉLIMINAIRES

- Annie Choquet-Geniet, 2100491474, *Les réseaux de Petri Un outil de modélisation*, Dunod, Paris, 2006 [ix-xii,1-11,101-106]
- Pierre Ladet, Outils de modélisation des automatismes séquentiels, Les réseaux de Petri, Technique de l'Ingénieur, n° R 7 252, 1989, [1-17]

CHAPITRE 2 : LES RÉSEAUX DE PETRI

- Annie Choquet-Geniet, 2100491474, *Les réseaux de Petri Un outil de modélisation*, Dunod, Paris, 2006 [13-98]
- Jean-Paul Leclercq, Cours de Théorie des Graphes et réseaux de Pétri, Université Notre-Dame de la Paix, Namur, 2010, [Chapitre 9]
- Stéphane Mariel, Vade-Mecum de l'étudiant en réseaux de Petri, <http://www.metz.supelec.fr/metz/personnel/vialle/course/CNAM-ACCOV-NFP103/extern-doc/RdP/vademecum-petri.pdf>, 2012, [1-6]
- G. Scorletti et G. Binet, Réseaux de Petri, Université de Caen/Basse-Normandie, 2006, [35-58]

CHAPITRE 3 : QUELQUES EXTENSIONS DES RÉSEAUX DE PETRI

RÉSEAU A ARC INHIBITEUR

- Annie Choquet-Geniet, 2100491474, *Les réseaux de Petri Un outil de modélisation*, Dunod, Paris, 2006 [125-129]
- Rodolphe Giroudeau, Cours sur l'Ingénierie des protocoles, Université de Montpellier, 2010 [48-49]
- Michel Combacau, Philippe Esteban, AlexandreNkeTsa, Commandes basées réseaux de Petri. Modélisation, analyse, Technique de l'Ingénieur, n°7 572, 2005, [6-7]

RÉSEAU SYNCHRONISÉ

- Annie Choquet-Geniet, 2100491474, *Les réseaux de Petri Un outil de modélisation*, Dunod, Paris, 2006 [152-156]
- G. Scorletti et G. Binet, Réseaux de Petri, Université de Caen/Basse-Normandie, 2006, [83-91]

RESEAU PREDICATS/TRANSITIONS

- Thomas Bourdeaud'huy, Apprentissage de comportement par synthèse de Réseaux de Petri, D.E.A d'automatique et d'informatique industrielle de Lille, 2001, 2012, <http://www3.ec-lille.fr/tomnab/recherche/dea/dea.pdf>, [57-58]
- Rodolphe Giroudeau, Cours sur l'Ingénierie des protocoles, Université de Montpellier, 2010 [78-79]
- Nabil Sadou, Thèse : Aide à la conception des systèmes embarqués sûrs de fonctionnement, Université de Toulouse III – Paul Sabatier, 2007, [60-61]
- Pascal Racloz, Didier Buchs, Réseaux de Petri : Introduction aux extensions, Université de Genève, 2009, 2012, https://smv.unige.ch/teaching/old-courses/Formal_tools_09_10/files/RdPExtIntro.pdf, [59-61]

RESEAU A FILES

- Annie Choquet-Geniet, 2100491474, *Les réseaux de Petri Un outil de modélisation*, Dunod, Paris, 2006 [129-134]

RESEAU DE PETRI À OBJET

- Rodolphe Giroudeau, Cours sur l'Ingénierie des protocoles, Université de Montpellier, 2010 [79-82]
- F. Raclot, D.Andreu, T. Libourel, R. Passama, E-NetObject : Un éditeur de Réseaux de Petri à Objets, Rapport LIRMM n° 02180 Université de Montpellier II, 2012, <http://hal.inria.fr/docs/00/19/15/13/PDF/D50.PDF> ,[5-7]

RESEAU DE PETRI FLOU

- Daniel Racocanu, Contribution à la surveillance des Systèmes de Production en utilisant les Techniques de l'Intelligence Artificielle, Université de Franche-Comté de Besançon, 2006,[157]
- Olivier Kaczor, Mémoire : Application d'algorithmes de bio-informatique à la recherche de patrons de conception, Université de Montréal, 2006, [20-21]
- Kevin Fong-Rey Liu, A fuzzy Petri net-based expert system and its application to damage assessment of bridges, National Central University, 1998, [47-48]
- M.H. Aziz, Erik L. J.Bohez, Manukid Parnichkun, Chanchal Saha, Classification of fuzzy Petri nets, and their applications, World Academy of Science, Engineering and Technology 72, 2010,[395-396, 494]
- H. Motameni, T. Ghassempouri, 1450223X Issue 16, Transforming fuzzy communication diagram to fuzzy Petri net, Amercian Journal of Scientific Research, 2011, [65-79]
- Matthieu Lescieux, Cours : Introduction à la logique floue. Application à la commande floue,2011, http://www.google.be/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CEoQFjAA&url=http%3A%2F%2Fauto.polytech.univ-tours.fr%2Fautomatique%2FAUA%2Fressources%2FIntroduction_logique_floue.ppt&ei=Pg8mUL7Wlc2KhQezmlDwBw&usg=AFQjCNE7rqLKDAzsq0kf4TYMxtmo80CdGA&sig2= Z8TLufzGkNfajIRqD-Pg, [1-38]
- Eugenia Minca, Florin Filip, Ryad Zemouri, Approche de la surveillance des systèmes par réseaux de Petri synchronisés flous, The publishing house of the romanian academy, Proceeding of the romanian academy, Series 1, Volume 9, Number 2/2008, 2008, [1-7]

CHAPITRE 4 : LES RÉSEAUX COLORÉS

- Annie Choquet-Geniet, 2100491474, *Les réseaux de Petri Un outil de modélisation*, Dunod, Paris, 2006 [107-114]
- Rodolphe Giroudeau, Cours sur l'Ingénierie des protocoles, Université de Montpellier, 2010 [65-76]
- Pascal Racloz, Didier Buchs, Présentation : Réseaux de Petri : Introduction aux extensions, Université de Genève, 2009, [13-58]
- G. Scorletti et G. Binet, Réseaux de Petri, Université de Caen/Basse-Normandie, 2006, [73-74]

CHAPITRE 5 : LES RÉSEAUX TEMPORISÉS

- Annie Choquet-Geniet, 2100491474, *Les réseaux de Petri Un outil de modélisation*, Dunod, Paris, 2006 [181-194]

- Rodolphe Giroudeau, Cours sur l'Ingénierie des protocoles, Université de Montpellier, 2010 [33-47]
- J.P Chemla, Cours de Modélisation et analyse des systèmes de production, Polytech'Tours, 2011, <http://auto.polytech.univ-tours.fr/automatique/ASP/ressources/ASP.pdf>, [13-15]
- Karen Godary, Cours d'automatisme et SED, RdP Les extensions, Polytech'Montpellier, 2008, 2011, http://www.lirmm.fr/~godary/Download/SED/Cours_RdP/CoursRdP_5_ExtensionRdP.pdf, [6-30]

CHAPITRE 6 : LES RÉSEAUX TEMPORELS

- Annie Choquet-Geniet, 2100491474, *Les réseaux de Petri Un outil de modélisation*, Dunod, Paris, 2006 [194-213]
- Rodolphe Giroudeau, Cours sur l'Ingénierie des protocoles, Université de Montpellier, 2010 [47-64]
- Florent Peres, Bernard Berthomieu, François Vernadat, Composer des réseaux de Petri temporels, Journal Européen des Systèmes Automatisé 43, 2009, [4-9]
- Bernard Berthomieu, La Méthode des Classes d'États pour l'Analyse des Réseaux Temporels, LAAS-CNRS, 2012, <http://projects.laas.fr/tina/papers/msr01.pdf>, [2-16]

CHAPITRE 7 : LES RÉSEAUX STOCHASTIQUES

- Rodolphe Giroudeau, Cours sur l'Ingénierie des protocoles, Université de Montpellier, 2010 [83-87]
- David Coupier, Processus Stochastique, Université Lille 1, 2012, <http://math.univ-lille1.fr/~coupier/Cours/CoursPS.pdf>, [3,8]
- J-P Lenoir, Probabilités-Statistiques : Principales distributions de probabilités, 2012, http://www.math.u-psud.fr/~pansu/web_ifips/Distr_Proba.pdf, [4,10]
- M.C Boukala, M.Ioualalen, Transformation sur les Réseaux de Petri Stochastiques, Application à l'évaluation des performances, LSI, Faculté d'Electronique et d'Informatique U.S.T.H.B, 2003, 2012, http://www.lsi-usthb.dz/Rapports_pdf/2003/LSI_TR_1003.pdf, [3]
- A. Abbas-Turki, O. Grunder, A. El Moudni, Modélisation et évaluation d'une correspondance élémentaire d'un système de transport en commun par réseau de Petri stochastiques, Laboratoire Système et Transport, UTBM, 2001, Troyes, [183-188]